

Колледж космического машиностроения и технологии

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей программного обеспечения для компьютерных систем

специальность 09.02.03 Программирование в компьютерных системах

Выполнили студенты:

Пихтов Виталий Алексеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Проверил преподаватель:

Гусятинер Леонид Борисович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2021

**Оглавление**

[Раздел 1. Техника решения задач с использованием структурного и объектно-ориентированного программирования. 3](#_Toc74932796)

[Практическое занятие №2 Установка интерпретатора Python 3 и настройка окружения. 3](#_Toc74932797)

[Практическое занятие №3.Техника работы в командной строке и среде IDLE. 5](#_Toc74932798)

[Практическое занятие №4.Техника работы с линейными и разветвляющимися программами. 9](#_Toc74932799)

[Практическое занятие №5.Техника работы с циклическими программами, цикл while. 14](#_Toc74932800)

[Практическое занятие №6. Техника работы с числами. 22](#_Toc74932801)

[Практическое занятие №7. Техника работы со строками. 24](#_Toc74932802)

[Практическое занятие №8. Техника работы со списками. 29](#_Toc74932803)

[Практическое занятие №9. Техника работы с циклом for и генераторами списков. 33](#_Toc74932804)

[Практическое занятие №10. Техника работы с функциями. 39](#_Toc74932805)

[Практическое занятие №11. Техника работы со словарями. 43](#_Toc74932806)

[Практическое занятие №12. Техника работы с множествами. 46](#_Toc74932807)

[Практическое занятие №13. Техника работы с кортежами. 49](#_Toc74932808)

[Практическое занятие №14. Техника работы с файлами. 52](#_Toc74932809)

[Практическое занятие №15. Техника работы с модулями. 53](#_Toc74932810)

[Практическое занятие №16. Техника работы с классами. 63](#_Toc74932811)

[**Раздел 2. Техника решения задач с использованием библиотек** 67](#_Toc74932812)

[Практическое занятие №17. Работа с ООП, классами. 67](#_Toc74932813)

[Практическое занятие №18. Работа с графикой и объектами в ней. 70](#_Toc74932814)

[Практическое занятие №19. Работа с сетевым подключением, сокетами, и многопоточностью. 113](#_Toc74932815)

[Практическое занятие №20. Работа с графами. 118](#_Toc74932816)

# Раздел 1. Техника решения задач с использованием структурного и объектно-ориентированного программирования.

# Практическое занятие №2 Установка интерпретатора Python 3 и настройка окружения.

Материал подготовил студент группы П2-18:

1| Пихтов Виталий Алексеевич

1.1 Установка интерпретатора Python 3 и настройка окружения

Для установки интерпретатора Python на компьютер, первое, что нужно сделать – это скачать дистрибутив. Загрузить его можно с официального сайта, перейдя по ссылке https://www.python.org/downloads/

Порядок установки на Windows:

1. Запустить скачанный установочный файл.

2. Выбрать способ установки.

3. Отметить необходимые опции установки (доступно при выборе Customize installation)

4. Выбираем место установки (доступно при выборе Customize installation)



Рис. 1 (Начало установки Python)



Рис. 2 (Настройка установки Python)



Рис. 3 (Настройка установки Python)



Рис. 4 (Завершение установки Python)

# Практическое занятие №3.Техника работы в командной строке и среде IDLE.

Выполняя (запуская) команду “python” в вашем терминале, вы получаете интерактивную оболочку Python.



Рис. 5 (Запуск Python в консоли)

Существует несколько способов закрыть оболочку Python:

>>> exit()

или же

>>> quit()

Кроме того, CTRL + D закроет оболочку и вернет вас в командную строку терминала. IDLE - простой редактор для Python, который поставляется вместе с Python. Откройте IDLE в вашей системе выбора. В оболочке есть подсказка из трех прямоугольных скобок:

>>>



Рис. 6 (Использование консоли IDLE Python)

Материал подготовил студент группы П2-18:

1| Пихтов Виталий Алексеевич  
Создание файла с кодом.

Код исходной программы:  
print('Hello, world') **(test.py)**

Запуск.

Для того, чтобы открыть существующий файл с кодом, нужно:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. рисунок 10)

3. Ввести команду “python название-файла”.

4. Получить результат выполнения программы (см. рисунок 11).

Import.

Чтобы подключить определенную библиотеку, нужно:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. рисунок 10)

3. Ввести в консоль команду “python”.

4. Ввести в консоль команду “import название-библиотеки”. (см. приложение 3)

Reload.

Чтобы перезагрузить определенное нечто, требуется:  
1. В папке с нужным файлом нажать сочетание клавиш Shift + ПКМ.

2. В появившемся диалоговом окне нажать “Открыть окно PowerShell здесь”. (см. рисунок 10)

3. Ввести в консоль команду “python”.  
4. Импортировать нужный файл. (см. рисунок 12)  
5. Перезаписать исходный файл.   
Код измененной программы:

print('Hello') **(test.py)**

6. Написать команду “reload sample”.

Отработка ошибок.

Примеры часто встречающихся ошибок:



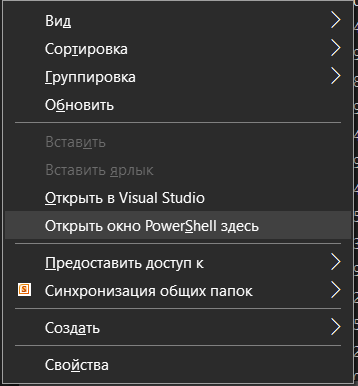
(рис. 7. Вывод ошибки)

Данная ошибка говорит о не прописанной команде “python”.



(рис. 8. Вывод ошибки)

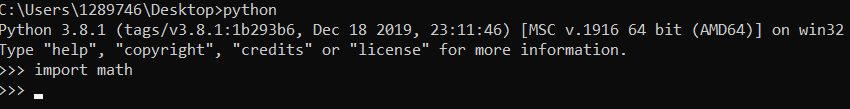
Данная ошибка говорит о том, что интерпретатор не может найти переменную с указанным именем.



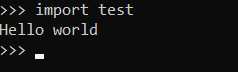
(рис. 9. Запуск консоли)



(рис. 10. Выполнение программы в консоли)



(рис. 11. Импорт библиотеки в консоли)



(рис. 12. Выполнение программы в консоли)

Составитель: Гусятинер Л.Б., 24.11.2020, МГОТУ ККМТ, П1-18, П2-18

Задание.

1. Показать работу в оболочке IDLE как в самой среде, так и путём запуска

файлов.

2. Показать умение работать с меню.



(Приложение 2. Работа с меню в IDLE)

# Практическое занятие №4.Техника работы с линейными и разветвляющимися программами.

Задачи на Boolean.

**Задача 1: ( begin1.py ). Листинг 1.**

Выполнил Пихтов В. А. П2-18.

Задача "Площадь прямоугольного треугольника".

Напишите программу, которая считывает длины двух катетов в прямоугольном треугольнике и выводит его площадь. Каждое число записано в отдельной строке.

a = int(input())

b = int(input())

print(a \* b / 2)

**Задача 2: ( begin2.py ). Листинг 2.**

Выполнил Пихтов В. А. П2-18.

Ввод и вывод данных. Задача «Дележ яблок».

n школьников делят k яблок поровну, неделящийся остаток остается в корзинке. Сколько яблок достанется каждому школьнику? Сколько яблок останется в корзинке? Программа получает на вход числа n и k и должна вывести искомое количество яблок (два числа).

n = int(input())

k = int(input())

print(k // n)

print(k % n)

Задачи на Case.

**Задача 1: ( case1.py ). Листинг 3.**

Выполнил Пихтов В. А. П2-18.

Пятилетний Петя учит цвета радуги. Напишите программу, которая принимает на вход натуральное число n и печатает первые n цветов радуги с большой буквы. При этом, если n > 7, программа должна ответить "Радуга состоит только из семи цветов".

RainbowSix = ("Красный", "Оранжевый", "Желтый", "Зеленый", "Голубой", "Синий", "Фиолетовый")

a = int(input())

if a <= 7:

for b in range(a):

print(RainbowSix[b])

elif a > 7:

print("Радуга состоит только из семи цветов")

**Задача 2: ( case2.py ). Листинг 4.**

Выполнил Пихтов В. А. П2-18.

Напишите программу, которая печатает названия месяцев до месяца n (n натуральное, вводится с клавиатуры) согласно шаблону, представленному в примере. Если n > 12, программа должна вывести "Error".

a = ["Первый месяц - январь", "Второй месяц - февраль", "Третий месяц - март", "Четвертый месяц - апрель", "Пятый месяц - май", "Шестой месяц - июнь", "Седьмой месяц - июль", "Восьмой месяц - август", "Девятый месяц - сентябрь", "Десятый месяц - октябрь", "Одиннадцатый месяц - ноябрь", "Двенадцатый месяц - декабрь",]

b = int(input())

if b <= 12:

for i in range(0, b):

print(a[i])

if b > 12:

print("Error")

**Задача 3: ( case3.py ) . Листинг 5.**

Выполнил Пихтов В. А. П2-18.

Некий гражданин NN пришел в магазин с n рублями. Просмотрев свой список покупок, он считает стоимость всех товаров и соотносит ее с той суммой, что у него есть. Если ему хватает денег, то выведите "Покупает", иначе "Не покупает".

Формат ввода: на первой строке вводится натуральное число – количество денег, на второй строке – количество товаров n, на следующих n строках – цена каждого из товаров.

Формат вывода: строки "Покупает" или "Не покупает".

money = int(input())

count = int(input())

counter = 0

for n in range(count):

a = int(input())

counter = counter + a

if counter <= money:

b = "Покупает"

else:

b = "Не покупает"

print(b)

Задачи на If.

**Задача 1: ( if1.py ) . Листинг 6.**

Выполнил Пихтов В. А. П2-18.

Напишите функцию fib(n), которая по данному целому неотрицательному n возвращает n-e число Фибоначчи. В этой задаче нельзя использовать циклы — используйте рекурсию.

def fib(n):

if (n == 1) or (n == 2):

return 1

else:

return fib(n - 1) + fib(n - 2)

n = int(input())

print(fib(n))

**Задача 2: ( if2.py ) . Листинг 7.**Выполнил Пихтов В. А. П2-18.

Выведите все четные элементы списка. При этом используйте цикл for, перебирающий элементы списка, а не их индексы!

spisok = input().split()

for n in spisok:

if int(n) % 2 == 0:

print(n, end = ' ')

**Задача 3: ( if3.py ) . Листинг 8.**

Выполнил Пихтов В. А. П2-18.

Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

spisok = [int(n) for n in input().split()]

for n in range(1, len(spisok)):

if spisok[n] > spisok[n - 1]:

print(spisok[n])

**Задача 1: ( boolean1.py ) . Листинг 9.**

Выполнил Пихтов В. А. П2-18.

Пользователь вводит числа по одному в строке, последовательность оканчивается числом 0. Выведите сумму чисел, введенных до нуля.

summa = 0

a = int(input())

while a != 0:

summa += a

a = int(input())

print(summa)

**Задача 2: ( boolean2.py ) . Листинг 10.**

Выполнил Пихтов В. А. П2-18.

Минимальный делитель.

Программа получает на вход натуральное число n > 1. Выведите минимальный делитель этого числа, отличный от единицы.

a = int(input())

n = 2

while a % n != 0:

n += 1

print(n)

Блок-схемы для задач к практическому занятию №4

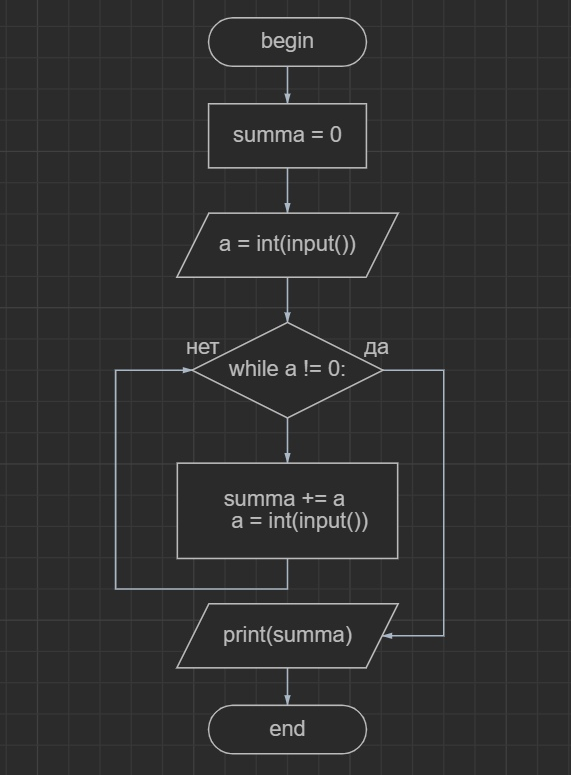


Рис. 13 (Блок-схема для задачи boolean1)

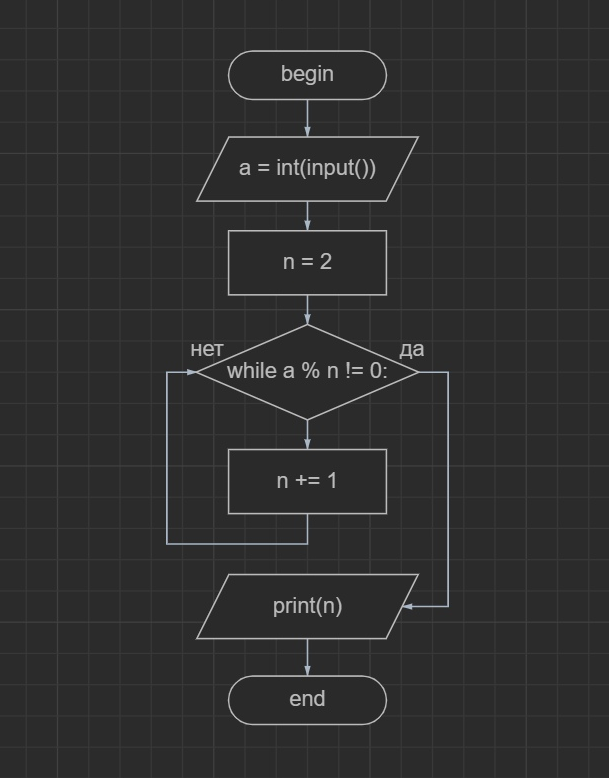


Рис. 14 (Блок-схема для задачи boolean2)

# Практическое занятие №5.Техника работы с циклическими программами, цикл while.

Задачи на For.

**Задача 1: ( for1.py ) . Листинг 11.**

Выполнил Пихтов В. А. П2-18.

Цикл с условием.

С клавиатуры вводятся числа a и b. Необходимо вывести те числа на полуинтервале [a; b) или [b; a), которые являются четными и дают остаток 1 при делении на 7.

Примечание: запись [a; b) означает промежуток из чисел от a включительно до b невключительно. В языке Python это соответствует записи range(a, b). Аналогично, промежуток [b; a) соответствует записи range(b, a).

a = int(input())

b = int(input())

if a < b:

for c in range(a, b):

if c % 7 == 1 and c % 2 == 0:

print(c)

else:

for c in range(b ,a):

if c % 7 == 1 and c % 2 == 0:

print(c)

**Задача 2: ( for2.py ) . Листинг 12.**

Выполнил Пихтов В. А. П2-18.

Посчитайте сумму всех чисел на полуинтервале [a; b) или [b; a), в зависимости от того, какое число больше.

a = int(input())

b = int(input())

sum = 0

if a < b:

for c in range(a, b):

sum += c

else:

for c in range(b, a):

sum += c

print(sum)

**Задача 3: ( for3.py ) . Листинг 13.**

Выполнил Пихтов В. А. П2-18.

Вы уже знакомы с термином "факториал", однако предыдущее наше решение не идеально – оно считает факториал только одного числа. Модифицируйте вашу программу из задачи "18!" таким образом, чтобы она принимала на вход натуральное число n и печатала значение выражения n!

a = int(input())

factorial = 1

for n in range(2, a+1):

factorial \*= n

print(factorial)

**Задача 4: ( for4.py ) . Листинг 14.**

Выполнил Пихтов В. А. П2-18.

Дан список из чисел и индекс элемента в списке k. Удалите из списка элемент с индексом k, сдвинув влево все элементы, стоящие правее элемента с индексом k.

Программа получает на вход список, затем число k. Программа сдвигает все элементы, а после этого удаляет последний элемент списка при помощи метода pop() без параметров.

Программа должна осуществлять сдвиг непосредственно в списке, а не делать это при выводе элементов. Также нельзя использовать дополнительный список. Также не следует использовать метод pop(k) с параметром.

spisok = [int(s) for s in input().split()]

a = int(input())

for n in range(a + 1, len(spisok)):

spisok[n - 1] = spisok[n]

spisok.pop()

print(' '.join([str(n) for n in spisok]))

**Задача 5: ( for5.py ) . Листинг 15.**

Выполнил Пихтов В. А. П2-18.

Выведите все элементы списка с четными индексами (то есть A[0], A[2], A[4], ...).

spisok = input().split()

for index in range(0, len(spisok), 2):

print(spisok[index])

Задачи на While.

**Задача 1: ( while1.py ) . Листинг 16.**

Выполнил Пихтов В. А. П2-18.

По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.

b = 1

a = int(input())

while b \*\* 2 <= a:

print(b \*\* 2)

b += 1

**Задача 2: ( while2.py ) . Листинг 17.**

Выполнил Пихтов В. А. П2-18.

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

b = 2

a = int(input())

while a % b != 0:

b += 1

print(b)

**Задача 3: ( while3.py ) . Листинг 18.**

Выполнил Пихтов В. А. П2-18.

По данному натуральному числу N найдите наибольшую целую степень двойки, не превосходящую N. Выведите показатель степени и саму степень.

Операцией возведения в степень пользоваться нельзя!

a = int(input())

b = 1

c = 2

while c <= a:

c \*= 2

b += 1

print(b - 1, c // 2)

**Задача 4: ( while4.py ) . Листинг 19.**

Выполнил Пихтов В. А. П2-18.

В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу y определите номер дня, на который пробег спортсмена составит не менее y километров.

Программа получает на вход действительные числа x и y и должна вывести одно натуральное число.

x = int(input())

y = int(input())

n = 1

while y > x:

x \*= 1.1

n += 1

print(n)

**Задача 5: ( while5.py ) . Листинг 20.**

Выполнил Пихтов В. А. П2-18.

Программа получает на вход последовательность целых неотрицательных чисел, каждое число записано в отдельной строке. Последовательность завершается числом 0, при считывании которого программа должна закончить свою работу и вывести количество членов последовательности (не считая завершающего числа 0). Числа, следующие за числом 0, считывать не нужно.

dlinna = 0

while int(input()) != 0:

dlinna += 1

print(dlinna)

Блок-схемы для задач к практическому занятию №5

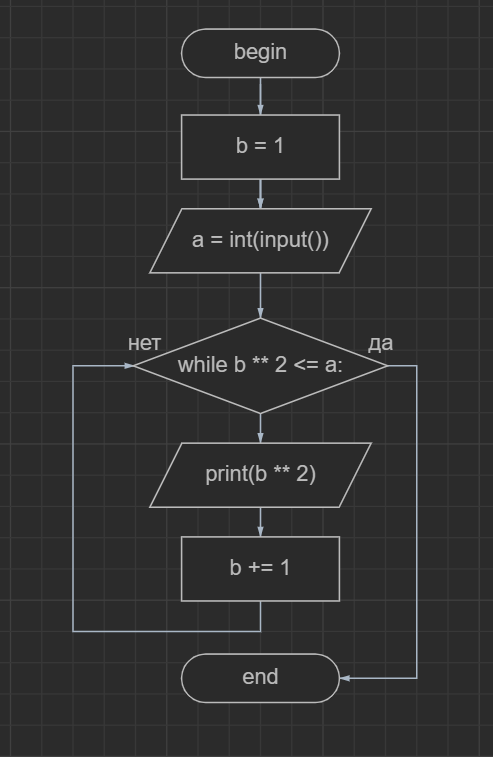


Рис. 15 (Блок-схема для задачи while1)

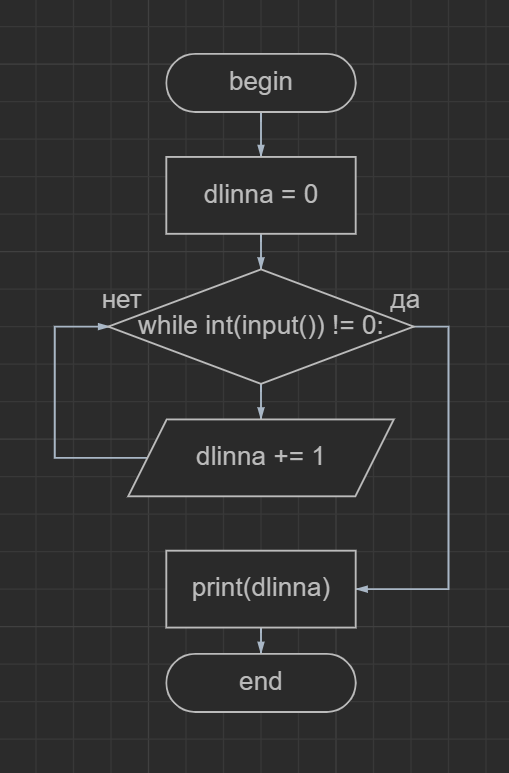


Рис. 16 (Блок-схема для задачи while5)

**Задание 2.** **Листинг 21.**

Выполнил: Сумин Константин Евгениевич

<https://stepik.org/lesson/3364/step/11?unit=947>

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу

в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

Sample Input 1:

5

-3

8

4

0

Sample Output 1:

14

(5.2-2.py)

mas=[-1]

print("Введите числа: ")

while(mas.count(0)!=1):

mas.insert(1,int(input()))

mas[0]=mas[0]+mas[1]

#print(mas) если нужно посмотреть лист

print("Сумма: ",mas[0]+1)

**Задание 3. Листинг 22.**

Выполнил: Сумин Константин ЕвгениевичРазработать программу для нахождения наибольшего общего делителя

(5.2-3.py)

print("Ввелите 2 числа: ")

a=int(input())

b=int(input())

if a<b:

a,b=b,a

for i in range (a,0,-1):

if a%i==0 and b%i==0:

break

print("Наибольший общий делитель: ",i)

**Задание 4. Листинг 23.**

Выполнил: Сумин Константин ЕвгениевичС использованием результата задания 2 разработать программу для нахождения наименьшего

общего кратного

(5.2-4.py)

mas=[-1]

print("Введите числа: ")

while(mas.count(0)!=1):

mas.insert(1,int(input()))

mas[0]=mas[0]+mas[1]

#print(mas) #если нужно посмотреть лист

print("Сумма: ",mas[0]+1)

c=1

for i in range (2,len(mas)-1,+1):

a=abs(mas[i])

b=abs(mas[i+1])

if a<b:

a,b=b,a

if a%b==0:

mas[i+1]=a

c=a

else:

c=a

mas[1]=0

while(mas[1]==0):

if c%b==0 and c%a==0:

mas[i+1]=c

mas[1]=1

else:

c+=1

print("НОК: ",c)

**Задание 5. Листинг 24.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/3369/step/8?unit=952>

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ...

(число повторяется столько раз, чему равно).

На вход программе передаётся неотрицательное целое число n — столько элементов

последовательности должна отобразить программа.

На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.

Sample Input:

7

Sample Output:

1 2 2 3 3 3 4

(5.2-5.py)

a = int(input())

b = []

for n in range(a):

j = 0

while j < n + 1:

b.append(n + 1)

j += 1

if len(b) > a: break

b = b[0:a]

for n in b:

print(n, end=" ")

# Практическое занятие №6. Техника работы с числами.

**Задачи Decimal. Листинг 25.**

Выполнил: Пихтов Виталий Алексеевич

Группа: П2-18

Decimal- это вычисления с заданной точностью

2.4 + 3.8 # Ответом будет 6.199999999999999, в нашем же случае 6.2

from decimal import Decimal

a = Decimal('2.4') + Decimal('3.8')

print(a)

2.4 + 3.8 == 6.2 # должно быть True, но с числами типа Decimal все верно:

c = Decimal('2.4') + Decimal('3.8')

print(c)

'''С помощью дополнительных знаков мы можем определить, сколько будет символов в дробной части числа:'''

from decimal import Decimal

num = Decimal("0.20")

num = 3 \* num

print(num) # 0.60

'''Однако нельзя смешивать в операциях дробные числа float и Decimal:'''

number = Decimal("0.1")

number = number + 0.1 # здесь возникнет ошибка

'''Округление чисел

Объекты Decimal имеют метод quantize(),который позволяет округлять числа.

В этот метод в качестве первого аргумента передается также объект Decimal,

который указывает формат округления числа:'''

number = Decimal("0.888")

number = number.quantize(Decimal("1.00"))

print(number) # 0.89

number = Decimal("0.1112345")

print(number.quantize(Decimal("1.00"))) # 0.11

number = Decimal("0.999")

print(number.quantize(Decimal("1.00"))) # 1.00

'''Как отделить целое от дроби в вещественных числах ?

Decimal вместо float:'''

a = Decimal('580.140')

b = int(a)

c = int(100 \* (a - b))

print(b, c) # => 580 14

(decimal1.py)

'''Использовался сайт

<https://metanit.com/python/tutorial/6.4.php> '''

Fraction.

'''Пихтов Виталий Алексеевич

Модуль fractions'''

from fractions import Fraction

from decimal import Decimal

import math

'''Fraction.limit\_denominator(max\_denominator=1000000)- ближайшее

рациональное число со знаменателем не больше данного.'''

a = Fraction(3.1415) ''' Целое и вещественное число, так же можно преобразовать в обыкновенную дробь'''

print(a) # 7074029114692207/2251799813685248

print(a.limit\_denominator()) # 6283/2000

# по умолчанию numerator = 0, denominator = 1

print(Fraction())

# равносильно Fraction(1, 2)

print(Fraction(numerator=1, denominator=2))

'''Если указанные числитель и знаменатель имеют общие делители,

то перед созданием рационального числа они будут сокращены'''

print(Fraction(8, 16), Fraction(15, 30))

'''Если указанные числитель и знаменатель имеют общие делители,

то перед созданием рационального числа они будут сокращены'''

print(Fraction(3, Fraction(1, 2)))

#Округляет до ближайшего четного числа.

print(Fraction('1/2').\_\_round\_\_())

'''создает обыкновенную дробь, которая является точным

представлением

десятичной дроби указанной в dec, где dec – это экземпляр класса decimal.Decimal'''

print(Fraction.from\_decimal(Decimal('0.7')))

'''принимает flt – число типа float и возвращает обыкновенную дробь отношение числителя

к знаменателю которой максимально приближается к значению flt.'''

print(Fraction.from\_float(0.5))

#fractions.gcd(a, b) - наибольший общий делитель чисел a и b.

print(math.gcd(1000, 3))

print(math.gcd(4, 6))

'''Использовался сайт <https://pyprog.pro/python/st_lib/fractions.html>

(fraction1.py)'''

# Практическое занятие №7. Техника работы со строками.

Техника работы со строками

**Задание 1. Листинг 26.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/201702/step/5?unit=175778>

С клавиатуры вводятся строки, последовательность заканчивается точкой.

Выведите буквы введенных слов в верхнем регистре, разделяя их пробелами.

(spiski1.py)

a = input()

List = []

while a != '.':

List.append(a)

a = input()

for n in List:

b = ''

for l in n:

b += l.upper() + ' '

print(b)

**Задание 2. Листинг 27.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/201702/step/8?unit=175778>

Известно, что для логина часто не разрешается использовать строки содержащие пробелы.

Но пользователю нашего сервиса особенно понравилась какая-то строка.

Замените пробелы в строке на символы нижнего подчеркивания, чтобы строка

могла сгодиться для логина. Если строка состоит из одного слова, менять ничего не нужно.

(spiski2.py)

a = input()

b = a.replace(" ", "\_")

print(b)

**Задание 3. Листинг 28.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/201702/step/9?unit=175778>

Уберите точки из введенного IP-адреса. Выведите сначала четыре числа через пробел, а затем сумму получившихся чисел.

(spiski3.py)

l = []

a = input().split(".")

for n in a:

l.append(int(n))

a = " ".join(a)

print(a, "\n", sum(l))

**Задание 4. Листинг 29.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/201702/step/14?unit=175778>

Программист логирует программу, чтобы хорошо знать, как она себя ведет (эта весьма распространенная и важная практика). Он использует разные типы сообщений для вывода ошибок (error), предупреждений (warning), информации (info) или подробного описания (verbose). Сообщения отличаются по внешнему виду. Назовем модификаторами такие символы, которые отличают сообщения друг от друга, позволяя программисту понять, к какому из типов относится сообщения. Модификаторы состоят из двух одинаковых символов и записываются по разу в начале и в конце строки.

@@ обозначает ошибку

!! обозначает предупреждение

// обозначает информационное сообщение

\*\* обозначает подробное сообщение

(spiski4.py)

a=input()

while a!='.':

if '@@' in a:

print('ошибка')

elif '!!' in a:

print('предупреждение')

elif '//' in a:

print('информация')

elif '\*\*' in a:

print('подробное сообщение')

a=input()

**Листинг 30.**

выполнил: Пихтов Виталий Алексеевич

1 Форматирование строк (оператор %)

Строки в Python содержат встроенную операцию, доступ к которой можно получить через оператор %. Это позволяет упростить позиционное форматирование. Вот простой пример:

Python

print('Hello, %s' % name)

# Вывод: "Hello, Bob"

print('Hello, %s' % name)

'''Вывод: "Hello, Bob"

Существуют другие определители формата, которые позволяют контролировать формат выдачи. Например, возможно конвертировать числа в шестнадцатеричную нотацию или добавлять пробелы для создания хорошо отформатированных таблиц и отчетов. (См Python Docs: ““printf-style String Formatting”.)

Можно использовать определитель формата %x для конвертации значения int в строку и представить его в качестве шестнадцатеричного числа:'''

Python

errno = 50159747054

print('%x' % errno)

# Вывод: 'badc0ffee'

errno = 50159747054

print('%x' % errno)

'''Вывод: 'badc0ffee'

Синтаксис форматирования строки немного меняется. Так как оператор % принимает только один аргумент, нужно обернуть правую часть в кортеж, вот так:'''

Python

errno = 50159747054

name = 'Bob'

print('Hey %s, there is a 0x%x error!' % (name, errno))

# 'Hey Bob, there is a 0xbadc0ffee error!'

errno = 50159747054

name = 'Bob'

print('Hey %s, there is a 0x%x error!' % (name, errno))

''''Hey Bob, there is a 0xbadc0ffee error!'

Также возможно сослаться на заменители переменных по имени в строке формата, если вы передадите сопоставление оператору %:'''

Python

print(

'Hey %(name)s, there is a 0x%(errno)x error!' % {

"name": name, "errno": errno

}

)

# Вывод: 'Hey Bob, there is a 0xbadc0ffee error!'

print(

'Hey %(name)s, there is a 0x%(errno)x error!' % {

"name": name, "errno": errno

}

)

'''Вывод: 'Hey Bob, there is a 0xbadc0ffee error!'

Это помогает сделать настройки формата более простым процессом, а также упрощает его модификацию в будущем. Недостаток — это то, что эта техника требует немного больше ручного ввода.

2 Форматирование строк (str.format)

Python 3 предоставил новый способ форматирования, который был внесен в раннюю версию Python 2.7. Это форматирование строк избавляется от специального синтаксиса оператора % и делает синтаксис для форматирования строк более регулярным. Теперь форматирование обрабатывается вызовом .format() в объекте строки.'''

Python

print('Hello, {}'.format(name))

# Вывод: 'Hello, Bob'

print('Hello, {}'.format(name))

'''Вывод: 'Hello, Bob'

Или можно сослаться на подстановки переменных по имени, и использовать их в том порядке, в котором хочется. Это достаточно мощный способ, так как он позволяет повторно упорядочить порядок отображения без изменения переданных функции format() аргументов:'''

Python

print(

'Hey {name}, there is a 0x{errno:x} error!'.format(

name=name, errno=errno

)

)

# Вывод: 'Hey Bob, there is a 0xbadc0ffee error!'

print(

'Hey {name}, there is a 0x{errno:x} error!'.format(

name=name, errno=errno

)

)

'''Вывод: 'Hey Bob, there is a 0xbadc0ffee error!'

Это также демонстрирует, что синтаксис формата переменной int — это шестнадцатеричная измененная строка.

Простой синтаксис

Синтаксис аналогичен тому, который вы используете в str.format(), но не такой перегруженный. Посмотрите на эту читабельность:'''

Python

name = "Eric"

age = 74

print(f"Hello, {name}. You are {age}.")

# Вывод: 'Hello, Eric. You are 74.'

name = "Eric"

age = 74

print(f"Hello, {name}. You are {age}.")

'''Вывод: 'Hello, Eric. You are 74.'

Произвольные выражения

Так как f-строки оцениваются по мере выражения, можно внести любую или все доступные выражения Python в них. Это позволит делать интересные вещи, например следующее:'''

Python

print(f"{2 \* 37}")

# Вывод: '74'

1

2

print(f"{2 \* 37}")

'''Вывод: '74'

Скорость

Буква f в f-strings может также означать и “fast”. f-строки заметно быстрее чем % и str.format() форматирования, f-строки являются выражениями, которые оцениваются по мере выполнения, а не постоянные значения.'''

# Практическое занятие №8. Техника работы со списками.

**Задание 1. Листинг 31.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/lists/problems/more_than_neighbours/>

Задача «Больше своих соседей»

Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух

своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда

не учитываются, поскольку у них недостаточно соседей (spisok1.py).

a = [int(n) for n in input().split()]

b = 0

for n in range(1, len(a) - 1):

if a[n - 1] < a[n] > a[n + 1]:

b += 1

print(b)

**Задание 2. Листинг 32.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/lists/problems/num_equal_pairs/>

Задача «Количество совпадающих пар»

Дан список чисел. Посчитайте, сколько в нем пар элементов, равных друг другу.

Считается, что любые два элемента, равные друг другу образуют одну пару, которую

необходимо посчитать (spisok2.py).

a = [int(n) for n in input().split()]

summa = 0

b = 0

while b != len(a) - 1:

for n in range(b + 1, len(a)):

if a[b] == a[n]:

summa += 1

b += 1

print(summa)

**Задание 3. Листинг 33.**

Выполнил: Сумин Константин Евгениевич

Составитель: Гусятинер Л.Б., 27.11.2020, МГОТУ ККМТ, П1-18, П2-18

Дано N списков целых чисел (N вводится с клавиатуры, сами списки заполняются

случайным образом). Требуется сформировать

- список, содержащий уникальные значения, попадающие в каждый из N списков

- список, содержащий уникальные значения, попадающие хотя бы в один из N списков

Решение без использования set - дополнительный бонус (spisok3.py).

from random import randint

li=[]

mas=[]

jojo=[]

jostar=[]

bug=[]

n = int(input("Введите количество списков: "))

temp=0

for i in range (0,n,+1):

for j in range(0,randint(2,5),+1):

a=randint(0,5)

mas.insert(j, a)

if li.count(a)==0:

li.append(a)

if bug.count(a)==0:

bug.append(a)

temp+=1

print("Список №",temp,": ",mas)

jojo.extend(bug)

bug.clear()

mas.clear()

for i in range(0,6,+1):

if jojo.count(i)==n:

jostar.append(i)

print("В каждом списке есть: ",jostar)

print("Хотя бы в одном списке есть: ",li)

**Задание 1. Array112. Листинг 34.**

выполнил: Пихтов Виталий Алексеевич

Дан массив A размера N.

Упорядочить его по возрастанию методом сортировки

простым обменом («пузырьковой» сортировкой):

просматривать массив, сравнивая его соседние элементы

(A0 и A1, A1 и A2 и т. д.) и меняя их местами,

если левый элемент пары больше правого; повторить описанные

действия N 1 раз. Для контроля за выполняемыми действиями

выводить содержимое массива после каждого просмотра.

Учесть, что при каждом просмотре количество анализируемых

пар можно уменьшить на 1.

import random

n = random.randrange(2,10)

a = [random.randrange(0,10) for i in range(n)]

print("N = ", n)

print("Array:")

print(a)

for i in range(1, n) :

for j in range(0,n - i) :

if a[j] > a[j + 1] :

a[j], a[j + 1] = a[j + 1], a[j]

print("Sort Array:")

print(a)

**Листинг 35.**

выполнил: Пихтов Виталий Алексеевич

Задание 2. Array113. Дан массив A размера N.

Упорядочить его по возрастанию методом сортировки простым

выбором: найти максимальный элемент массива и поменять его

местами с последним (N-1 м) элементом; выполнить описанные

действия N 1 раз, каждый раз уменьшая на 1 количество

анализируемых элементов и выводя содержимое массива.

import random

n = random.randrange(2, 10)

a = [random.randrange(1, 8) for i in range(n)]

print(n)

print(a)

for i in range(1, n):

d = False

b = a[i]

c = i - 1

while c >= 0 :

if b >= a[c] :

break

else :

a[c + 1] = a[c]

d = True

c -= 1

if d and c >= -1:

a[c + 1] = b

print(i)

print(a)

print(a)

**Задание 3. Array114. Листинг 36.**

выполнил: Пихтов Виталий Алексеевич

Дан массив A размера N. Упорядочить

его по возрастанию методом сортировки простыми вставками:

сравнить элементы A0 и A1 и, при необходимости меняя их

местами, добиться того, чтобы они оказались упорядоченными

по возрастанию; затем обратиться к элементу A2 и

переместить его в левую (уже упорядоченную) часть массива,

сохранив ее упорядоченность; повторить этот процесс для

остальных элементов, выводя содержимое массива после

обработки каждого элемента (от 1-го до N-1 го).

import random

n = random.randrange(2, 10)

a = [random.randrange(1, 8) for i in range(n)]

print(n)

print(a)

for i in range(1, n):

d = False

b = a[i]

c = i - 1

while c >= 0 :

if b >= a[c] :

break

else :

a[c + 1] = a[c]

d = True

c -= 1

if d and c >= -1:

a[c + 1] = b

print(i)

print(a)

print(a)

# Практическое занятие №9. Техника работы с циклом for и генераторами списков.

**Задание 1. Листинг 37.**

Выполнил: Сумин Константин Евгениевич

Составитель: Гусятинер Л.Б., 27.11.2020, МГОТУ ККМТ, П1-18, П2-18

Для проведения конкурса проектов в ККМТ формируются группы

из 4х участников: coder, writer, tester, designer, программирующих

на одном и том же языке.

Каждый студент может программировать только на одном языке

и занимать только одну позицию.

Дан текстовый файл, содержащий перечень студентов с указанием языка и позиции

(каждый студент с новой строки)

Требуется

1. Получить список студентов с указанием языка и позиции

2. Сформировать список всевозможных команд

3. Вывести список команд с указанием состава и названия команды:

Команда 1

coder: ...

designer: ...

tester: ...

writer: ...

Команда 2

...

Пункты 1 и 2 выполнить с использованием генераторов списка (kkmt.py)

kkmt=[]

kod=[]

file = open('name.txt', encoding='utf-8')

name=list(tuple(item.replace('\n', '') for item in line.split(' ')) for line in file)

for i in range(0,len(name),+1):

if kod.count(name[i][1]) == 0:

kod.append(name[i][1])

for i in range(0,len(kod),+1):

for j in range(0,len(name),+1):

temp=0

while temp<3:

if name[j][1] == kod[i] and temp==0 and name[j][2=='coder'] and kkmt.count(name[j])!=name.count(name[j]):

kkmt.append(name[j])

temp+=1

elif name[j][1] == kod[i] and temp==1 and name[j][2=='designer'] and kkmt.count(name[j])!=name.count(name[j]):

kkmt.append(name[j])

temp += 1

elif name[j][1] == kod[i] and temp==2 and name[j][2=='tester'] and kkmt.count(name[j])!=name.count(name[j]):

kkmt.append(name[j])

temp += 1

elif name[j][1] == kod[i] and temp==3 and name[j][2=='writer'] and kkmt.count(name[j])!=name.count(name[j]):

kkmt.append(name[j])

temp += 1

else:

temp=4

try:

n=0

pr=0

bug=0

aaa=0

for j in range(0,len(kod),+1):

coun=1

temp=0

for k in range (0,len(kkmt),+1):

temp+=kkmt[k].count(kod[j])

if temp/4!=0:

coun=int(temp/4)

if coun==1:

for i in range(n,n+temp,+1):

if bug%4==0:

print("Команда №",int(pr/4)+1)

print(kkmt[i])

pr += 1

aaa += 1

if bug!=3:

bug+=1

else:

bug=0

else:

for i in range(n, n + temp, +1):

if bug%4==0 and i%2!=0:

print("Команда №",int(pr/4)+1)

if i%2!=0:

print(kkmt[i])

pr += 1

aaa += 0.5

if bug != 3:

bug += 1

else:

bug = 0

for i in range(n, n + temp, +1):

if bug%4==0 and i%2==0:

print("Команда №",int(pr/4)+1)

if i%2==0:

print(kkmt[i])

pr+=1

aaa += 0.5

if bug != 3:

bug += 1

else:

bug = 0

n+=temp

if aaa!=len(kod)\*4:

print("Всязи с недостатком участников, некоторые команды были сформированы некорректно\n"

"Пожалуйста, уберите участников которым не хватает людей для команды, или найдите\n"

"им людей в команду")

except:

print("Всязи с недостатком участников, некоторые команды были сформированы некорректно\n"

"Пожалуйста, уберите участников которым не хватает людей для команды, или найдите\n"

"им людей в команду")

**Задачи одномерные массивы.**

**Задание 1. Array55. Листинг 38.**

<http://ptaskbook.com/ru/tasks/array.php>

выполнил: Пихтов Виталий Алексеевич

Дан целочисленный массив A размера N (<= 15). Переписать в новый целочисленный

массив B все элементы с нечетными порядковыми номерами (1, 3, ...) и вывести размер

полученного массива B и его содержимое. Условный оператор не использовать. (array55.py)

import random

n = random.randrange(2, 24)

a = [i for i in range(n)]

b = a[1::2]

print(n)

print(a)

print(len(b))

print(b)

**Задание 2. Array57. Листинг 39.**

выполнил: Пихтов Виталий Алексеевич

Дан целочисленный массив A размера N. Переписать в новый целочисленный массив B

того же размера вначале все элементы исходного массива с четными номерами,

а затем — с нечетными:

A[0], A[2], A[4], A[6], ..., A[1], A[3], A[5], ... .

Условный оператор не использовать. (array57.py)

import random

n = random.randrange(2, 22)

a = [i for i in range(n)]

b = a[0::2] + a[1::2]

print(n)

print(a)

print(len(b))

print(b)

**Задание 3. Array58. Листинг 40.**

выполнил: Пихтов Виталий Алексеевич

Дан массив A размера N. Сформировать новый массив B того же размера по

следующему правилу: элемент B[K] равен сумме элементов массива A с номерами от 0 до K. (array58.py)

import random

n = random.randrange(2,22)

a = [i for i in range(n)]

b = []

b.append(a[0])

print(n)

print(a)

for i in range(1, n) :

b.append(a[i] + b[i - 1])

print(len(b))

print(b)

**Задачи двумерные массивы (матрицы).**

**Задание 4. Matrix3. Листинг 41.**

<http://ptaskbook.com/ru/tasks/matrix.php>

выполнил: Пихтов Виталий Алексеевич

Даны целые положительные числа M, N и набор из M чисел. Сформировать

матрицу размера M x N, у которой в каждом столбце содержатся все числа из исходного

набора (в том же порядке). (matrix3.py)

import random

import numpy

a = random.randrange(2,10)

b = random.randrange(2,10)

print("a: ", a, "b: ", b)

c = numpy.zeros((a, b))

d = [random.randrange(1,5) for i in range(a)]

print(a)

print(d)

for i in range(a):

for j in range(b):

c[i][j] = d[i]

print(c)

matrix = []

for i in range(a):

matrix.append([d[i] for j in range(b)])

print(matrix)

**Задание 5. Matrix56. Листинг 42.**

Выполнил: Сумин Константин Евгениевич

Дана матрица размера M x N (N — четное число). Поменять местами

левую и правую половины матрицы. (matrix56.py)

matrix = [[1, 3, 5, 0], [3, 1, 8, 7], [9, 0, 6, 7]]

def swap(m: list):

size = int(len(m[0]) / 2)

left\_half = []

right\_half = []

for s in m:

left\_half.append(list(s[i] for i in range(size)))

right\_half.append(list(s[i] for i in range(size, len(s))))

res = []

for i in range(len(m)):

res.append(right\_half[i] + left\_half[i])

return res

for string in matrix:

for item in string:

print(item, end=' ')

print()

print()

result = swap(matrix)

for string in result:

for item in string:

print(item, end=' ')

print()

**Задание 6. Matrix88. Листинг 43.**

Выполнил: Сумин Константин Евгениевич

Дана квадратная матрица порядка M. Обнулить элементы матрицы,

лежащие ниже главной диагонали. Условный оператор не использовать. (matrix88.py)

matrix = [[1, 3, 5, 0], [3, 1, 8, 7], [9, 0, 6, 7], [0, 1, 7, 3]]

def m\_z(m: list):

for i in range(len(m)):

for j in range(i):

m[i][j] = 0

for string in matrix:

for item in string:

print(item, end=' ')

print()

print()

m\_z(matrix)

for string in matrix:

for item in string:

print(item, end=' ')

print()

print()

# Практическое занятие №10. Техника работы с функциями.

**Задачи на функции.**

**Задание 2. Листинг 44.**

выполнил: Сумин Константин Евгениевич

Описать функцию SumRange(A, B) целого типа, находящую сумму всех целых

чисел от A до B включительно (A и B — целые). Если A > B, то функция возвращает 0.

С помощью этой функции найти суммы чисел от A до B и от B до C, если даны числа A, B, C.

def Why(c):

try:

int(c)

return True

except:

return False

def SumRange(a:int,b:int,c:str):

if Why(c):

j=0

for i in range (a,b+1,+1):

j+=i

print("От А до B: ",j)

j = 0

if int(c)>b:

for i in range(b, int(c)+1, +1):

j += i

else:

for i in range(int(c), b + 1, +1):

j += i

print("От B до C: ", j)

else:

j=0

for i in range (a,b+1,+1):

j+=i

print("От А до B: ", j)

a=int(input("A:"))

b=int(input("B:"))

print("Введите в С букву, если не хотите добовлять С")

c=input("C:")

if a>b:

print('0')

raise SystemExit

SumRange(a,b,c)

**Задание 3. Func10. Листинг 45.**

выполнил: Сумин Константин Евгениевич

Описать функцию IsSquare(K) логического типа, возвращающую True,

если целый параметр K (> 0) является квадратом некоторого целого числа, и False

в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых

положительных чисел.

import random

from math import sqrt

def IsSquare(K:int):

if int(sqrt(k))==sqrt(k):

return True

return False

k=int(input("K: "))

print(IsSquare(k))

print("Квадраты 10 случайных чисел:")

for i in range (10):

k=int(random.randint(0,100))

print(k,":",IsSquare(k))

**Задание 4. Func33. Листинг 46.**

выполнил: Сумин Константин Евгениевич

Описать функцию SortInc3(X), меняющую содержимое списка X из трех

вещественных элементов таким образом, чтобы их значения оказались упорядоченными по

возрастанию (функция возвращает None). С помощью этой функции упорядочить по

возрастанию два данных списка X и Y.

import random

def SortInc3(x:list):

print("Исходный список:",'\n',x)

for i in range (2):

for j in range (1,3,+1):

if x[i]>x[j]:

x[i],x[j]=x[j],x[i]

print("Отсортированный список:", '\n', x)

return None

for i in range (2):

x=[random.randint(0,100),random.randint(0,100),random.randint(0,100)]

SortInc3(x)

**Задание 2. Листинг 47.**

выполнил: Сумин Константин Евгениевич

<https://stepik.org/lesson/201702/step/13?unit=175778>

Использовать map, lambda

Квадраты в обратном порядке. Числа вводятся до точки. Через пробел выведите эти числа в

обратном порядке, возводя их в квадрат.

Sample Input:

5

16

20

1

9

.

Sample Output:

81 1 400 256 25

a=[]

try:

while True:

x=input()

a.append(int(x))

except:

x=x

b=list(map(lambda x: x\*x,a))

c=[]

for i in range (len(b)-1,-1,-1):

c.append(b[i])

print(' '.join(map(str, c)))

**Задание 3. Листинг 48.**

выполнил: Сумин Константин Евгениевич

Использовать lambda, filter.

Array55. Дан целочисленный массив A размера N (<= 15). Переписать в новый целочисленный

массив B все элементы с нечетными порядковыми номерами (1, 3, ...) и вывести размер

полученного массива B и его содержимое. Условный оператор не использовать.

n=int(input("Введите количество чисел в массиве "))

a=[]

b=[]

for i in range (n):

a.append(int(input()))

for i in range(0,n,+2): #Если счет начинать с 0, то заменить на range(1,n,+2)

b.append(a[i])

print("Размер массива: ",len(b),"\nСодержимое: ",b)

**Задание 4. Листинг 49.**

выполнил: Сумин Константин Евгениевич

Использовать lambda, map.

<https://stepik.org/lesson/239422/step/2?unit=211833>

Быстрая инициализация. Программа получает на вход три числа через пробел — начало и конец

диапазона, а также степень, в которую нужно возвести каждое число из диапазона. Выведите

числа получившегося списка через пробел.

Sample Input:

5 12 3

Sample Output:

125 216 343 512 729 1000 1331 1728

a=[]

b=[]

x={0:'Введите начало диапазона',1:'Введите конец диапазона',2:'Введите степень'}

for i in range (3):

print(x[i])

a.append(int(input()))

if a[0]<a[1]:

for i in range(a[0],a[1]+1,+1):

b.append(i a[2])

else:

for i in range(a[0], a[1]-1, -1):

b.append(i a[2])

print(' '.join((map(str,b))))

# Практическое занятие №11. Техника работы со словарями.

**Задание 1. Листинг 50.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/dicts/problems/occurency_index/>

Задача «Номер появления слова»

Условие. В единственной строке записан текст. Для каждого слова из данного текста

подсчитайте, сколько раз оно встречалось в этом тексте ранее.

Словом считается последовательность непробельных символов идущих подряд, слова разделены

одним или большим числом пробелов или символами конца строки. (stroki1.py)

a = dict()

for c in input().split():

a[c] = a.get(c, 0) + 1

print(a[c] - 1, end=' ')

**Задание 2. Листинг 51.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/dicts/problems/permissions/>

Задача «Права доступа»

Условие. В файловую систему одного суперкомпьютера проник вирус, который сломал контроль

за правами доступа к файлам. Для каждого файла известно, с какими действиями можно к

нему обращаться:

запись W,

чтение R,

запуск X.

В первой строке содержится число N — количество файлов содержащихся в данной файловой

системе. В следующих N строчках содержатся имена файлов и допустимых с ними операций,

разделенные пробелами. Далее указано чиcло M — количество запросов к файлам. В последних

M строках указан запрос вида Операция Файл. К одному и тому же файлу может быть применено

любое колличество запросов.

Вам требуется восстановить контроль над правами доступа к файлам (ваша программа для

каждого запроса должна будет возвращать OK если над файлом выполняется допустимая

операция, или же Access denied, если операция недопустима. (stroki2.py)

a = [input().split() for i in range(int(input()))]

b = dict()

for i in a:

for j in range(1, len(i)):

if i[j] == 'R':

b[i[0]] = b.get(i[0], ['']) + ['read']

elif i[j] == 'W':

b[i[0]] = b.get(i[0], ['']) + ['write']

else:

b[i[0]] = b.get(i[0], ['']) + ['execute']

for i in range(int(input())):

act, name = input().split()

if act in b[name]:

print('OK')

else:

print('Access denied')

**Задание 3. Листинг 52.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/dicts/problems/most_frequent_word/>

Задача «Самое частое слово»

Условие. Дан текст: в первой строке задано число строк, далее идут сами строки.

Выведите слово, которое в этом тексте встречается чаще всего. Если таких слов несколько,

выведите то, которое меньше в лексикографическом порядке. (stroki3.py)

a = int(input())

b = {}

max = 0

key = ''

for i in range(a):

S = input().split()

for j in S:

b[j] = b.get(j, 0) + 1

for k, v in sorted(b.items()):

if v > max:

max = v

key = k

print(key)

**Задание 1. Листинг 53.**

Выполнил: Сумин Константин Евгениевич

<https://stepik.org/lesson/243394/step/4?unit=215740>

Телефонная книга. Этап 1. Коля устал запоминать телефонные номера и заказал у Вас

программу, которая заменила бы ему телефонную книгу. Коля может послать программе

два вида запросов: строку, содержащую имя контакта и его номер, разделенные пробелом,

или просто имя контакта. В первом случае программа должна добавить в книгу новый номер,

во втором – вывести номер контакта. Ввод происходит до символа точки. Если введенное

имя уже содержится в списке контактов, необходимо перезаписать номер.

Sample Input:

Ben 89001234050

Alice 210-220

Alice

Alice 404-502

Nick +16507811251

Ben

Alex +4(908)273-22-42

Alice

Nick

Robert 51234047129

Alex

.

Sample Output:

210-220

89001234050

404-502

+16507811251

+4(908)273-22-42

contacts={}

a=0

while a!='.':

a=input()

b=a.split(' ')

if a!='.':

if len(b)==2:

contacts[b[0]]=b[1]

elif len(b)==1:

print(contacts[b[0]])

# Практическое занятие №12. Техника работы с множествами.

**Задание 1. Листинг 54.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/sets/problems/number_of_unique/>

Задача «Количество различных чисел»

Условие. Дан список чисел. Определите, сколько в нем встречается различных чисел. (spisok1.py)

spisok = input().split()

mnogestvo = set(spisok)

dlina = len(mnogestvo)

print(dlina)

**Задание 2. Листинг 55.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/sets/problems/number_of_coincidental/>

Задача «Количество совпадающих чисел»

Условие. Даны два списка чисел. Посчитайте, сколько чисел содержится одновременно как

в первом списке, так и во втором.(spisok2.py)

a = {int(s) for s in input().split()}

b = {int(s) for s in input().split()}

c = a & b

print(len(c))

**Задание 3. Листинг 56.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/sets/problems/sets_intersection/>

Задача «Пересечение множеств»

Условие. Даны два списка чисел. Найдите все числа, которые входят как в первый,

так и во второй список и выведите их в порядке возрастания.(spisok3.py)

a = [int(x) for x in input().split()]

b = [int(x) for x in input().split()]

c = [str(x) for x in sorted(list(set(a) & set (b)))]

d = ' '.join(c)

print(d)

**Задание 4. Листинг 57.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/sets/problems/number_of_words/>

Задача «Количество слов в тексте»

Условие. Дан текст: в первой строке записано число строк, далее идут сами строки.

Определите, сколько различных слов содержится в этом тексте.

Словом считается последовательность непробельных символов идущих подряд, слова разделены

одним или большим числом пробелов или символами конца строки.(spisok4.py)

a = set()

for n in range(int(input())):

a.update(input().split())

print(len(a))

**Задание 5. Листинг 58.**

выполнил: Пихтов Виталий Алексеевич

<https://pythontutor.ru/lessons/sets/problems/polyglotes/>

Задача «Полиглоты»

Условие. Каждый из некоторого множества школьников некоторой школы знает некоторое

количество языков. Нужно определить сколько языков знают все школьники, и сколько языков

знает хотя бы один из школьников.

В первой строке задано количество школьников. Для каждого из школьников сперва записано

количество языков, которое он знает, а затем - названия языков, по одному в строке.

В первой строке выведите количество языков, которые знают все школьники. Начиная со

второй строки - список таких языков. Затем - количество языков, которые знает хотя бы

один школьник, на следующих строках - список таких языков. Языки нужно выводить в

лексикографическом порядке, по одному на строке.''' (spisok5.py)

all = int(input())

a = set()

sum = set()

b = set()

for i in range(all):

a = int(input())

for i in range(a):

b.update(input().split())

sum |= b

a = sum & b

b = set()

print(len(a))

print(\*sorted(a))

**Задание 1. Листинг 59.**

Выполнил: Сумин Константин Евгениевич

<https://stepik.org/lesson/3380/step/3?unit=963>

Простейшая система проверки орфографии может быть основана на использовании списка известных слов.

Если введённое слово не найдено в этом списке, оно помечается как "ошибка".

Попробуем написать подобную систему.

На вход программе первой строкой передаётся количество d известных нам слов, после чего

на d строках указываются эти слова.

Затем передаётся количество l строк текста для проверки, после чего l строк текста.

Выведите уникальные "ошибки" в произвольном порядке. Работу производите без учёта регистра.

Sample Input:

4

champions

we

are

Stepik

3

We are the champignons

We Are The Champions

Stepic

Sample Output:

stepic

champignons

the'''

di=[]

er=[]

n=int(input("Введите количество словарных слов: "))

print("Введите словарные слова")

for i in range (n):

lit=str(input())

di.append(lit.upper())

l=int(input("Введите количество строк для проверки: "))

print("Введите строки для проверки")

for i in range (l):

litr=str(input())

test=0

for j in range (len(litr)):

if litr[j]==' ':

if di.count(litr[test:j].upper())==0 and er.count(litr[test:j].upper())==0:

er.append(litr[test:j].upper())

test=j+1

if j==len(litr)-1:

if di.count(litr[test:j+1].upper())==0 and er.count(litr[test:j].upper())==0:

er.append(litr[test:j+1].upper())

print(er)

# Практическое занятие №13. Техника работы с кортежами.

**Задание 1. Листинг 60.**

выполнил: Пихтов Виталий Алексеевич

<https://stepik.org/lesson/193753/step/4?unit=168148>

Вывести чётные

Необходимо вывести все четные числа на отрезке [a; a \* 10].

Sample Input:

2

Sample Output:

(2, 4, 6, 8, 10, 12, 14, 16, 18, 20)

a = int(input())

start = a if a % 2 == 0 else a + 1

end = a \* 10 + 1

k = tuple(range(start, end, 2))

print(k)

**Задание 2. Листинг 61.**

Выполнил: Сумин Константин Евгениевич

<https://stepik.org/lesson/193753/step/5?unit=168148>

Убывающий ряд.

С клавиатуры вводятся целые числа a > b. Выведите убывающую последовательность чисел

по одному числу в строке.

Sample Input:

-2

-8

Sample Output:

-2

-3

-4

-5

-6

-7

a = int(input())

b = int(input())

k = tuple(range(a, b, -1))

print(\*k, sep='\n')

**Задание 3. Листинг 62.**

Выполнил: Сумин Константин Евгениевич

Составитель: Гусятинер Л.Б., 24.11.2020, МГОТУ ККМТ, П1-18, П2-18

В каждой строке файла хранится информация о пунктах и их координатах

относительно некоторого центра.

Требуется

1. Прочесть файл в список кортежей

2. Найти диаметр множества точек, то есть расстояние между наиболее удалёнными точками.

Указать наиболее удалённые пары

3. Сформировать список пар городов, имеющих одинаковое расстояние до центра

4. Отсортировать список одним из методов, реализованных в предыдущих работах

Результаты вывести на экран

Пример входного файла

Москва 0 0

Ивантеевка 20 15

Щёлково 10 30

Пушкино 15 5

from math import sqrt

def abc(s:str):

try:

int(s)

return True

except ValueError:

return False

def distance(x1, y1, x2, y2):

dx = x1 - x2 if x1 > x2 else x2 - x1

dy = y1 - y2 if y1 > y2 else y2 - y1

return sqrt(dx \* dx + dy \* dy)

def distancee(x1,y1):

return sqrt(x1\*x1+y1\*y1)

'''1 Прочесть файл в список кортежей'''

li= {}

f = open('goroda.txt', encoding='utf-8')

cities = list(tuple(int(item.replace('\n', '')) if abc(item.replace('\n', ''))

else item.replace('\n', '') for item in line.split(' ')) for line in f)

'''2 Найти диаметр множества точек, то есть расстояние между наиболее удалёнными точками.

Указать наиболее удалённые пары'''

for i in range(0,len(cities)-1,+1):

x1=cities[i][1]

y1=cities[i][2]

for j in range(i+1, len(cities), +1):

x2=cities[j][1]

y2=cities[j][2]

if li.get(distance(x1, y1, x2, y2)) is None:

li[distance(x1, y1, x2, y2)] = cities[i][0] + "-" + cities[j][0]

else:

li[distance(x1, y1, x2, y2)] += ' '+cities[i][0]+"-"+cities[j][0]

max = 0

for k in li.keys():

if max < k:

max = k

for l in li.keys():

if max == l:

print(li[l])

'''3 Сформировать список пар городов, имеющих одинаковое расстояние до центра'''

lii={}

for i in range(0,len(cities),+1):

x1=cities[i][1]

y1=cities[i][2]

if lii.get(distancee(x1, y1)) is None:

lii[distancee(x1, y1)] = cities[i][0]

else:

lii[distancee(x1, y1)] += ' '+cities[i][0]

print(lii.values())

'''4 Отсортировать список одним из методов, реализованных в предыдущих работах

Результаты вывести на экран'''

liii=[]

for j in range(0,len(cities),+1):

max = 0

for i in range(0,len(cities),+1):

x1=cities[i][1]

y1=cities[i][2]

if distancee(x1, y1)>max and liii.count(distancee(x1, y1))== 0:

max=distancee(x1, y1)

if liii.count(max)== 0:

liii.append(max)

print("От большего к меньшему",liii)

Класс namedtuple() модуля collections возвращает новый подкласс кортежа с именем typename. Новый подкласс используется для создания объектов, похожих на кортежи, которые имеют индексируемые и итерируемые поля, доступные для поиска по атрибутам. Экземпляры подкласса также имеют полезную строку документации с typename и field\_names, а так же метод \_\_repr\_\_(), который перечисляет содержимое кортежа в формате name=value.

Пример:

from collections import namedtuple

Book = namedtuple('Book', ['id', 'title', 'authors'])

Book.\_\_doc\_\_ += ': Hardcover book in active collection'

Book.id.\_\_doc\_\_ = '13-digit ISBN'

Book.title.\_\_doc\_\_ = 'Title of first printing'

Book.authors.\_\_doc\_\_ = 'List of authors sorted by last name'

# Практическое занятие №14. Техника работы с файлами.

**Задание 1. Листинг 63.**

выполнил: Пихтов Виталий Алексеевич

<http://ptaskbook.com/ru/tasks/text.php>

Text5. Дана строка S и текстовый файл. Добавить строку S в конец файла.

v = open('text.txt','a');

v.write('S');

**Задание 3. Листинг 64.**

выполнил: Сумин Константин Евгениевич

<http://ptaskbook.com/ru/tasks/text.php>

Text20. Дан текстовый файл. Заменить в нем все подряд идущие пробелы на один пробел.

with open('space.txt', 'r',encoding='utf-8') as f:

fil = f.read()

while True:

if fil.count(' ')!=0:

fil = fil.replace(' ', ' ')

else:

break

with open('space.txt', 'w',encoding='utf-8') as f:

f.write(fil)

# Практическое занятие №15. Техника работы с модулями.

**Листинг 65.**

Составитель: Гусятинер Л.Б., 3.12.2020, МГОТУ ККМТ, П1-18, П2-18

Выполнил: Пихтов Виталий Алексеевич

collections.deque(iterable, [maxlen]) - создаёт очередь из итерируемого объекта с максимальной длиной maxlen. Очереди очень похожи на списки, за исключением того, что добавлять и удалять элементы можно либо справа, либо слева.

Методы, определённые в deque:

append(x) - добавляет x в конец.

appendleft(x) - добавляет x в начало.

clear() - очищает очередь.

count(x) - количество элементов, равных x.

extend(iterable) - добавляет в конец все элементы iterable.

extendleft(iterable) - добавляет в начало все элементы iterable (начиная с последнего элемента iterable).

pop() - удаляет и возвращает последний элемент очереди.

popleft() - удаляет и возвращает первый элемент очереди.

remove(value) - удаляет первое вхождение value.

reverse() - разворачивает очередь.

rotate(n) - последовательно переносит n элементов из начала в конец (если n отрицательно, то с конца в начало).

collections.Counter

collections.Counter - вид словаря, который позволяет нам считать количество неизменяемых объектов (в большинстве случаев, строк). Пример:

import collections

c = collections.Counter()

for word in ['spam', 'egg', 'spam', 'counter', 'counter', 'counter']:

... c[word] += 1

...

print(c)

Counter({'counter': 3, 'spam': 2, 'egg': 1})

print(c['counter'])

# 3

print(c['collections'])

# 0

Но возможности Counter на этом не заканчиваются. У него есть несколько специальных методов:

elements() - возвращает список элементов в лексикографическом порядке.

c = Counter(a=4, b=2, c=0, d=-2)

list(c.elements())

['a', 'a', 'a', 'a', 'b', 'b']

most\_common([n]) - возвращает n наиболее часто встречающихся элементов, в порядке убывания встречаемости. Если n не указано, возвращаются все элементы.

Counter('abracadabra').most\_common(3)

[('a', 5), ('r', 2), ('b', 2)]

subtract([iterable-or-mapping]) # - вычитание

c = Counter(a=4, b=2, c=0, d=-2)

d = Counter(a=1, b=2, c=3, d=4)

c.subtract(d)

Counter({'a': 3, 'b': 0, 'c': -3, 'd': -6})

Наиболее часто употребляемые шаблоны для работы с Counter:

sum(c.values()) # - общее количество.

c.clear() # - очистить счётчик.

list(c) # - список уникальных элементов.

set(c) # - преобразовать в множество.

dict(c) # - преобразовать в словарь.

c.most\_common()[:-n:-1] # - n наименее часто встречающихся элементов.

c += Counter() # - удалить элементы, встречающиеся менее одного раза.

Counter также поддерживает сложение, вычитание, пересечение и объединение:

c = Counter(a=3, b=1)

d = Counter(a=1, b=2)

c + d

Counter({'a': 4, 'b': 3})

c - d

Counter({'a': 2})

c & d

Counter({'a': 1, 'b': 1})

c | d

Counter({'a': 3, 'b': 2})

**Листинг 66.**

Выполнил: Пихтов Виталий Алексеевич

collections.defaultdict ничем не отличается от обычного словаря за исключением того, что по умолчанию всегда вызывается функция, возвращающая значение:

import collections

defdict = collections.defaultdict(list)

print(defdict)

defaultdict(<class 'list'>, {})

for i in range(5):

... defdict[i].append(i)

...

print(defdict)

defaultdict(<class 'list'>, {0: [0], 1: [1], 2: [2], 3: [3], 4: [4]})

collections.OrderedDict

collections.OrderedDict - ещё один похожий на словарь объект, но он помнит порядок, в котором ему были даны ключи. Методы:

popitem(last=True) - удаляет последний элемент если last=True, и первый, если last=False.

move\_to\_end(key, last=True) - добавляет ключ в конец если last=True, и в начало, если last=False.

d = {'banana': 3, 'apple':4, 'pear': 1, 'orange': 2}

OrderedDict(sorted(d.items(), key=lambda t: t[0]))

OrderedDict([('apple', 4), ('banana', 3), ('orange', 2), ('pear', 1)])

OrderedDict(sorted(d.items(), key=lambda t: t[1]))

OrderedDict([('pear', 1), ('orange', 2), ('banana', 3), ('apple', 4)])

OrderedDict(sorted(d.items(), key=lambda t: len(t[0])))

OrderedDict([('pear', 1), ('apple', 4), ('orange', 2), ('banana', 3)])

**Листинг 67.**

выполнил: Пихтов Виталий Алексеевич

Метод argv возвращает аргументы командной строки, переданные скрипту Python, в виде списка. Важно отметить, что первый аргумент (с индексом 0) в списке — это название самого скрипта. Остальные представлены в виде последовательности.В следующем примере файл с кодом называется test.py.

import sys

print("Привет {}. Добро пожаловать в руководство по {} на {}".format(sys.argv[1], sys.argv[2], sys.argv[3]))

Запуск test.py в командной строке вместе со следующими аргументами:

python test.py Студенты sys PythonRu

В указанной выше команде первым аргументом функции argv является элемент Студенты. sys и PythonRu — второй и третий аргументы.

Вывод:

Привет Студенты. Добро пожаловать в руководство по sys на PythonRu

Имя используемой OS

Модуль os предоставляет множество функций для работы с операционной системой, причём их поведение, как правило, не зависит от ОС, поэтому программы остаются переносимыми. Здесь будут приведены наиболее часто используемые из них.

Будьте внимательны: некоторые функции из этого модуля поддерживаются не всеми ОС.

os.name - имя операционной системы. Доступные варианты: 'posix', 'nt', 'mac', 'os2', 'ce', 'java'.

import os

print(os.name) # ответ: nt

os.environ - словарь переменных окружения. Изменяемый (можно добавлять и удалять переменные окружения).

import os

print(os.environ)

os.getlogin() - имя пользователя, вошедшего в терминал (Unix).

os.getpid() - текущий id процесса.

os.uname() - информация об ОС. возвращает объект с атрибутами: sysname - имя операционной системы, nodename - имя машины в сети (определяется реализацией), release - релиз, version - версия, machine - идентификатор машины.

os.access(path, mode, \*, dir\_fd=None, effective\_ids=False, follow\_symlinks=True) - проверка доступа к объекту у текущего пользователя. Флаги: os.F\_OK - объект существует, os.R\_OK - доступен на чтение, os.W\_OK - доступен на запись, os.X\_OK - доступен на исполнение.

os.chdir(path) - смена текущей директории.

os.chmod(path, mode, \*, dir\_fd=None, follow\_symlinks=True) - смена прав доступа к объекту (mode - восьмеричное число).

os.chown(path, uid, gid, \*, dir\_fd=None, follow\_symlinks=True) - меняет id владельца и группы (Unix).

os.getcwd() - текущая рабочая директория.

os.link(src, dst, \*, src\_dir\_fd=None, dst\_dir\_fd=None, follow\_symlinks=True) - создаёт жёсткую ссылку.

os.listdir(path=".") - список файлов и директорий в папке.

os.mkdir(path, mode=0o777, \*, dir\_fd=None) - создаёт директорию. OSError, если директория существует.

os.makedirs(path, mode=0o777, exist\_ok=False) - создаёт директорию, создавая при этом промежуточные директории.

os.remove(path, \*, dir\_fd=None) - удаляет путь к файлу.

os.rename(src, dst, \*, src\_dir\_fd=None, dst\_dir\_fd=None) - переименовывает файл или директорию из src в dst.

os.renames(old, new) - переименовывает old в new, создавая промежуточные директории.

os.replace(src, dst, \*, src\_dir\_fd=None, dst\_dir\_fd=None) - переименовывает из src в dst с принудительной заменой.

os.rmdir(path, \*, dir\_fd=None) - удаляет пустую директорию.

os.removedirs(path) - удаляет директорию, затем пытается удалить родительские директории, и удаляет их рекурсивно, пока они пусты.

os.symlink(source, link\_name, target\_is\_directory=False, \*, dir\_fd=None) - создаёт символическую ссылку на объект.

os.sync() - записывает все данные на диск (Unix).

os.truncate(path, length) - обрезает файл до длины length.

os.utime(path, times=None, \*, ns=None, dir\_fd=None, follow\_symlinks=True) - модификация времени последнего доступа и изменения файла. Либо times - кортеж (время доступа в секундах, время изменения в секундах), либо ns - кортеж (время доступа в наносекундах, время изменения в наносекундах).

os.walk(top, topdown=True, onerror=None, followlinks=False) - генерация имён файлов в дереве каталогов, сверху вниз (если topdown равен True), либо снизу вверх (если False). Для каждого каталога функция walk возвращает кортеж (путь к каталогу, список каталогов, список файлов).

os.system(command) - исполняет системную команду, возвращает код её завершения (в случае успеха 0).

os.urandom(n) - n случайных байт. Возможно использование этой функции в криптографических целях.

os.path - модуль, реализующий некоторые полезные функции на работы с путями.

Различные сведения о версии Python

sys.copyright

Строка, дающая информацию об авторских правах на интерпретатор Python.

sys.version

Строка, состоящая из номера версии Python, а также дополнительной информации о номере сборки и используемом компиляторе.

Например:

3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit

(Intel)]

sys.api\_version

Информацию об используемой интерпретатором версии C API. Применяется при отладке конфликтов, возникающих при взаимодействии Python и дополнительных модулей.

Пример значения: 1013.

sys.version\_info:

Переменная sys.api\_version возвращает кортеж, содержащий пять компонентов номера версии:

* major,
* minor,
* micro,
* releaselevel,
* serial.

import sys

sys.version\_info.major

sys.implementation

Объект, дающий информацию о запущенном в данный момент интерпретаторе Python.

Объект должен содержать атрибуты: имя, версия, версия в шестнадцатеричном представлении, дескриптор кэша.

Пример:

namespace(cache\_tag='cpython-38', hexversion=50856176, name='cpython',

version=sys.version\_info(major=3, minor=8, micro=0, releaselevel='final',

serial=0))

sys.hexversion

Номер версии интерпретатора Python, закодированный одним числом. Это число увеличивается с каждой версией, включая все виды релизов.

sys.hexversion используют, чтобы удостовериться, поддерживает ли интерпретатор какую-либо функцию. Если нет, то функция заменяется каким-либо поддерживаемым аналогом.

Это число называется «hexversion», потому что оно принимает понятный вид, только если его передать в функцию hex().

Пример версии: 50856176.

sys.winver

Содержит номер версии Python, который используется в реестре Windows. Например, «3.8-32».

Используется только в информационных целях, то есть даже если изменить значения «sys.winver», значения в реестре Windows не изменятся.

Каталоги и пути интерпретатора Python.

sys.prefix

Атрибут sys.prefix это строка, предоставляет специфичный для площадки префикс каталога, в котором установлены независимые от платформы файлы Python. По умолчанию, это строка '/usr/local'.

Значение sys.prefix устанавливается во время сборки Python с помощью аргумента --prefix для скрипта configure, например $ ./configure --prefix=/opt/python-3.x.x/". Основная коллекция модулей библиотеки Python установлена в каталоге prefix/lib/pythonX.Y, а независимые от платформы заголовочные файлы (все, кроме pyconfig.h) хранятся в prefix/include/pythonX.Y, где XY - номер версии Python, например 3.7.

import sys

sys.prefix

sys.real\_prefix

sys.base\_prefix

Атрибут sys.base\_prefix устанавливается при запуске Python перед запуском site.py то же значение, что и атрибут sys.prefix. Если не работает в виртуальной среде, значения останутся прежними. Если site.py обнаружит, что используется виртуальная среда, то значения sys.prefix и sys.exec\_prefix будут изменены на значения установки виртуальной среды, тогда как sys.base\_prefix и sys.base\_exec\_prefix останутся указывать на базовую установку Python (ту, из которой была создана виртуальная среда).

sys.exec\_prefix

Атрибут sys.exec\_prefix это строка, задает префикс каталога для конкретной программы, где установлены зависящие от платформы файлы Python. По умолчанию это '/usr/local'.

Значение sys.prefix устанавливается во время сборки Python с помощью аргумента --exec-prefix для скрипта configure. В частности, все файлы конфигурации (например, заголовочный файл pyconfig.h) установлены в каталоге exec\_prefix/lib/pythonX.Y/config, а модули разделяемой библиотеки установлены в exec\_prefix/lib/pythonX.Y/lib-dynload, где XY номер версии Python, например 3.7.

sys.base\_exec\_prefix:

Атрибут sys.exec\_prefix устанавливает при запуске Python до запуска site.py то же значение, что и sys.exec\_prefix. Если программа не работает в виртуальной среде, то значения останутся прежними. Если site.py обнаружит, что используется виртуальная среда, то значения sys.prefix и sys.exec\_prefix будут изменены на значения установки виртуальной среды, тогда как sys.base\_prefix и sys.base\_exec\_prefix останутся указывать на базовую установку Python (ту, из которой была создана виртуальная среда).

sys.executable:

Атрибут sys.executable это строка, задающая абсолютный путь исполняемого двоичного файла для интерпретатора Python в системах, где это имеет смысл. Если Python не может получить реальный путь к своему исполняемому файлу, sys.executable будет пустой строкой или None.

sys.platlibdir:

Атрибут sys.exec\_prefix это имя каталога библиотеки для конкретной платформы. Он используется для построения пути к стандартной библиотеке и путей установленных модулей расширения. (Новое в Python 3.9.)

На большинстве платформ он равен lib. В Fedora и SuSE на 64-битных платформах он равен lib64, что дает следующие пути sys.path (где X.Y - версия Python major.minor):

* /usr/lib64/pythonX.Y/: Стандартная библиотека (например, os.py модуля os),
* /usr/lib64/pythonX.Y/lib-dynload/: модули расширения языка C стандартной библиотеки (например, модуль errno, точное имя файла зависит от платформы),
* /usr/lib/pythonX.Y/site-packages/ (всегда используйте lib, а не sys.platlibdir): сторонние модули,
* /usr/lib64/pythonX.Y/site-packages/: модули расширения языка C сторонних пакетов.

Объекты stdin, stdout, stderr модуля sys в Python.

Файловые объекты интерпретатора для стандартного ввода, вывода.

Синтаксис:

import sys

sys.stdin

sys.stdout

sys.stderr

Описание:

Файловые объекты sys.stdin, sys.stdout и sys.stderr используются интерпретатором для стандартного ввода, вывода и ошибок:

* sys.stdin - используется для всех интерактивных входных данных, включая вызовы input();
* sys.stdout - используется для вывода оператором print() и выражений, которые возвращают значение, а также для подсказок input();
* sys.stderr - сообщения об ошибках и собственные запросы переводчика.

Примечание. При некоторых условиях sys.stdin, sys.stderr и sys.stdout, а также исходные значения sys.\_\_stdin\_\_, sys.\_\_stdout\_\_ и sys.\_\_stderr\_\_ могут быть None. Обычно это относится к приложениям с графическим интерфейсом Windows, которые не подключены к консоли, а приложения Python запускаются с pythonw.exe(1).

Примеры использования sys.stdout, sys.stdin:

import sys

for line in sys.stdin:

print(line.rstrip('\n')[::-1])

''''''''''''''''''''''''''''''''''''''''''''''''

import sys

sys.stdout.write("Hello, ")

sys.stdout.write("World!")

# Hello, World!

Функция exit() модуля sys в Python.

Быстрый выход из программы при возникновении ошибки.

Синтаксис:

import sys

sys.exit([arg])

Параметры:

arg - статус выхода, по умолчанию 0

Возвращаемое значение:

целое число, указывающее статус выхода.

Описание:

Функция exit() модуля sys - выход из Python. Она реализуется путем вызова исключения SystemExit, поэтому выполняются действия по очистке, указанные в предложениях finally операторов try и можно перехватить попытку выхода на внешнем уровне.

Пример:

ref = osgeo.ogr.Open(reference)

if ref is None:

raise SystemExit('Unable to open %s' % reference)

или:

ref = osgeo.ogr.Open(reference)

if ref is None:

print('Unable to open %s' % reference)

sys.exit(-1)

# Практическое занятие №16. Техника работы с классами.

**Классы и объекты Python. Листинг 68.**

выполнил: Пихтов Виталий Алексеевич

<https://pythonru.com/uroki/klassy-i-obekty-python-uroki-po-python-dlja-nachinajushhih>

Задание 1. Создание классов

Чтобы определить новый класс в своей программе, необходимо напечатать ключевое слово class, а после него добавить имя для создаваемой структуры данных, завершив ввод вставкой двоеточия. Следующий пример демонстрирует генерацию пустого класса с именем Example. Как можно заметить, в нем полностью отсутствует какая-либо информация.

class Example:

pass

example = Example()

Несмотря на пустое тело класса Example, на его основе уже можно создать определенный объект, обладающий уникальным идентификатором. Последняя строка кода, находящегося выше, представляет собой пример генерации объекта с именем example и типом данных Example. Здесь используется оператор присваивания, а также пустые круглые скобки после названия класса, прямо как в вызове метода не имеющего никаких аргументов.

Определив новый класс, можно создавать сколько угодно объектов на его основе. Как уже было сказано выше, такая структура данных может включать в себя некие свойства, то есть переменные, которыми будет наделен каждый экземпляр класса. Ниже приведен простой пример класса и объекта Python 3. В примере описывается класс под названием Data со строкой word и числом number.

class Data:

word = "Python"

number = 3

data = Data()

print(data.word + " " + str(data.number))

Задание 2. Создание объекта

Python 3

Если создать объект, основанный на классе Data, то он получит обе переменные, а также их значения, которые были определены изначально. Таким образом, был сгенерирован объект data. Получить доступ к его полям с именами word и number можно с помощью оператора точки, вызвав его через экземпляр класса. Функция print поможет вывести значения полей объекта data на экран. Не стоит забывать и о том, что число следует привести к строчному виду для того чтобы обработать его в методе print вместе с текстовым значением.

Помимо полей, пользовательский класс может включать в себя и методы, которыми будут наделены все его экземпляры. Вызвать выполнение определенного метода через созданный объект можно так же, как и получить доступ к его полям, то есть с помощью точки. Данный пример демонстрирует класс Data с функцией sayHello, которая выводит текст на экран.

class Data:

def sayHello(self):

print("Hello World!")

data = Data()

data.sayHello()

Hello World!

Задание 3. Функция init

В объектно-ориентированном программировании конструктором класса называют метод, который автоматически вызывается при создании объектов. Его также можно назвать конструктором объектов класса. Имя такого метода обычно регламентируется синтаксисом конкретного языка программирования. Так в Java имя конструктора класса совпадает с именем самого класса. В Python же роль конструктора играет метод \_\_init\_\_().

В Python наличие пар знаков подчеркивания спереди и сзади в имени метода говорит о том, что он принадлежит к группе методов перегрузки операторов. Если подобные методы определены в классе, то объекты могут участвовать в таких операциях как сложение, вычитание, вызываться как функции и др.

При этом методы перегрузки операторов не надо вызывать по имени. Вызовом для них является сам факт участия объекта в определенной операции. В случае конструктора класса – это операция создания объекта. Так как объект создается в момент вызова класса по имени, то в этот момент вызывается метод \_\_init\_\_(), если он определен в классе.

Необходимость конструкторов связана с тем, что нередко объекты должны иметь собственные свойства сразу. Пусть имеется класс Person, объекты которого обязательно должны иметь имя и фамилию. Если класс будет описан подобным образом

class Person:

def setName(self, n, s):

self.name = n

self.surname = s

то создание объекта возможно без полей. Для установки имени и фамилии метод setName() нужно вызывать отдельно:

>>> from test import Person

>>> p1 = Person()

>>> p1.setName("Bill", "Ross")

>>> p1.name, p1.surname

('Bill', 'Ross')

В свою очередь, конструктор класса не позволит создать объект без обязательных полей:

class Person:

def \_\_init\_\_(self, n, s):

self.name = n

self.surname = s

p1 = Person("Sam", "Baker")

print(p1.name, p1.surname)

Задание 4. Методы объектов

Возможность создания классовых методов является одним из следствий того, что в Python классы также являются объектами. Для создания классового (только «новые» классы могут иметь классовые методы) метода можно использовать декоратор classmethod

>>> class A(object):

def \_\_init\_\_(self, int\_val):

self.val = int\_val + 1

@classmethod

def fromString(cls, val): # вместо self принято использовать cls

return cls(int(val))

...

>>> class B(A):pass

...

>>> x = A.fromString("1")

>>> print x.\_\_class\_\_.\_\_name\_\_

A

>>> x = B.fromString("1")

>>> print x.\_\_class\_\_.\_\_name\_\_

B

Задание 5. Параметр self

В методах класса первый параметр функции по соглашению называют self, и это ссылка на сам объект этого класса. Но это соглашение. Можно называть параметры как угодно.

Есть объект нужно создать/изменить поле у этого объекта из метода этого самого объекта.

class Human:

def \_\_init\_\_(self):

self.blood = 7000

def add\_blood(self, volume):

self.blood += volume

Задание 6. Изменение свойств объекта

В Python объекту можно не только переопределять поля и методы, унаследованные от класса, также можно добавить новые, которых нет в классе:

>>> l.test = "hi"

>>> B.test

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

AttributeError: type object 'B' has no

attribute 'test'

>>> l.test

'hi'

Задание 7. Удалить свойства объекта

Свойства объектов можно удалять с помощью ключевого слова del

del p1.age

Задание 8. Удаление объектов

удалить объекты, используя ключевое слово del.

del p1

# **Раздел 2. Техника решения задач с использованием библиотек**

# Практическое занятие №17. Работа с ООП, классами.

**Листинг 68.**

Работу выполнил: Виталий Пихтов А.

Паттерн наблюдатель с использованием библиотеки векторов, меню, и функциями сложения всех чисел в массиве и вывод чисел Фибоначчи с num элементов.

#include <iostream>

#include <vector>

class EventObserver

{

public:

virtual void Handle(int num) = 0;

};

class Event

{

public:

std::vector<EventObserver\*> observer;

void notify(int num)

{

for (EventObserver\* o : observer)

{

o->Handle(num);

}

}

};

class Menu

{

protected:

int num, num2;

public:

Event sum;

Event fib;

int count;

void setSum(int value)

{

num = value;

if (num > 0)

{

sum.notify(num);

}

else

{

std::cout << "Error" << std::endl;

}

}

int getSum()

{

count = num;

return num;

}

void giveFib(int value)

{

num2 = value;

fib.notify(num2);

}

};

class EventSum : public EventObserver //  класс наследующий класс наблюдателя, и выводящий сумму всех чисел в массиве

{

protected:

int numArray = 0, arrayNum[100];

public:

void Handle(int num) override

{

std::cout << "Enter " << num << " num: " << std::endl;

for (int i = 0; i < num; i++)

{

std::cin >> arrayNum[i];

}

for (int i = 0; i < num; i++)

{

numArray += arrayNum[i];

}

std::cout << "Sum: " << numArray << std::endl;

}

};

class EventFib : public EventObserver //  класс наследующий класс наблюдателя, и выводящий числа фибоначчи

{

protected:

int arrayNum[100];

public:

void Handle(int num) override

{

std::cout << "Enter " << num << " num: " << std::endl;

arrayNum[0] = 0;

arrayNum[1] = 1;

std::cout << arrayNum[0] << "\n" << arrayNum[1] << std::endl;

for (int i = 0; i < num; i++)

{

if (i >= 1)

{

arrayNum[i + 1] = arrayNum[i - 1] + arrayNum[i];

std::cout << "Sum: " << arrayNum[i + 1] << std::endl;

}

}

}

};

int main()

{

int num = 0, option = 0;

bool isOpen = true;

Menu choise;

std::cout << "Enter count num: " << std::endl;

std::cin >> num;

choise.sum.observer.push\_back(new EventSum());

choise.fib.observer.push\_back(new EventFib());

while (isOpen)

{

std::cout << std::endl;

std::cout << "/////////////////" << std::endl;

std::cout << "Enter operation: " << std::endl;

std::cout << "//1. sum " << std::endl;

std::cout << "//2. fib " << std::endl;

std::cout << "//3. exit " << std::endl;

std::cout << std::endl;

std::cin >> option;

switch (option)

{

case 1:

choise.setSum(num);

break;

case 2:

choise.giveFib(num);

break;

case 3:

isOpen = false;

break;

default:

break;

}

}

return 0;

}

# Практическое занятие №18. Работа с графикой и объектами в ней.

**Листинг 69.**

Выполнил: Пихтов Виталий Алексеевич

Приложение 2D-платформер. Выполняет отрисовку всей графики и звука через библиотеку SFML, игровые динамические объекты написаны в классах (игрок, противники, зелья здоровья), карты столкновений (имеют различные свойства такие как столкновение, перемещение игрока в другую точку карты, отнимают у игрока и противников все жизни) читаются из файлов (level1.txt, level2.txt, level3.txt) и вводятся в двумерный массив из которого считывается местоположение препятствий. Текстуры статических объектов накладываются отдельно от карт столкновений. Использовались библиотеки SFML, WINDOWS, FSTREAM, MATH, STRING, CHRONO, THREAD.

#include <iostream>

#include "SFML/Graphics.hpp"

#include "SFML/Audio.hpp"

#include <fstream>

#include <SFML/Window.hpp>

#include <Windows.h>

#include <math.h>

#include <thread>

#include <chrono>

#include <string>

using namespace sf;

using namespace std;

int ground = 400;

const int H = 128;

const int W = 366;

const int sizeMapCollisionA = 16;

const int sizeMapCollisionB = 16;

const int sizeMapCollisionL = 16;

const int sizeMapCollisionI = 16;

int choiseCharacterGame;

String level;

String TileMap[H];

String TileMap2[H];

String TileMap3[H];

// Функция чтения карты с файла в массив

void map1()

{

std::ifstream ifs("level1.txt");

std::string content((std::istreambuf\_iterator<char>(ifs)), (std::istreambuf\_iterator<char>()));

int i = 0;

String line = "";

for (char c : content)

{

if (c != '\n')

{

line += c;

}

else

{

TileMap[i] = line;

line = "";

i++;

}

}

}

// Функция чтения карты с файла в массив

void map2()

{

std::ifstream ifs("level2.txt");

std::string content((std::istreambuf\_iterator<char>(ifs)), (std::istreambuf\_iterator<char>()));

int i = 0;

String line = "";

for (char c : content)

{

if (c != '\n')

{

line += c;

}

else

{

TileMap2[i] = line;

line = "";

i++;

}

}

}

// Функция чтения карты с файла в массив

void map3()

{

std::ifstream ifs("level3.txt");

std::string content((std::istreambuf\_iterator<char>(ifs)), (std::istreambuf\_iterator<char>()));

int i = 0;

String line = "";

for (char c : content)

{

if (c != '\n')

{

line += c;

}

else

{

TileMap3[i] = line;

line = "";

i++;

}

}

}

////////////////////////////////////////////

// Первый враг с индивидуальными параметрами

class ENEMY {

public:

float dx, dy, hp = 100;

FloatRect rect, rect2;

bool onGround, enemyWalkLadderCollision = false, ladderWalk = false, attack = false, attack2 = false, death = false, destroy = false, animationDamage = false, shield = false, level2 = false;

int left;

Sprite sprite;

float CurrentFrameRun, CurrentFrameAttack, CurrentFrameIdle, CurrentFrameDamage, CurrentFrameDeath, CurrentFrameShield, stamina = 100;

ENEMY(Texture& image)

{

sprite.setTexture(image);

rect = FloatRect(600, 580, 44, 52);

dx = dy = 0;

CurrentFrameRun = 0;

CurrentFrameAttack = 0;

CurrentFrameIdle = 0;

CurrentFrameDamage = 0;

CurrentFrameDeath = 0;

CurrentFrameShield = 0;

}

////////////////////////////////////////////////////////////////////////

// Функция проверки столкновения с объектами в двумерном массиве TileMap

void Collision(int num)

{

for (int i = rect.top / sizeMapCollisionA; i < (rect.top + rect.height) / sizeMapCollisionA; i++)

for (int j = rect.left / sizeMapCollisionA; j < (rect.left + rect.width) / sizeMapCollisionA; j++)

{

// Проверка на размер карты массива

if ((j < 0) or (i < 0) or (j > 366) or (i > 128))

{

return;

}

// Стена без свойств

if (level2 == false)

{

if (TileMap[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

// невидимая стена для врагов

if (TileMap[i][j] == 'J')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

}

// Если идет переход на другой уровень то считывается другая карта массивов для столкновения

if (level2 == true)

{

// Стена без свойств

if (TileMap2[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

// Невидимая стена для врагов

if (TileMap2[i][j] == 'J')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

}

}

}

// Функция срабатывающая каждый кадр

void update(float time)

{

float ground = 1768;

rect.left += dx \* time;

Collision(0);

if (!onGround) dy = dy + 0.0005 \* time;

rect.top += dy \* time;

onGround = false;

Collision(1);

if (rect.top > ground)

{

rect.top = ground;

dy = 0;

onGround = true;

};

// Скорость проигрывания анимаций

CurrentFrameRun += 0.12;

CurrentFrameIdle += 0.12;

CurrentFrameAttack += 0.12;

CurrentFrameDamage += 0.13;

CurrentFrameDeath += 0.12;

CurrentFrameShield += 0.12;

// Колличество кадров анимаций

if (CurrentFrameRun > 6)

CurrentFrameRun -= 6;

if (CurrentFrameIdle > 4)

CurrentFrameIdle -= 4;

if (CurrentFrameAttack > 12)

CurrentFrameAttack -= 12;

if (CurrentFrameDamage > 3)

CurrentFrameDamage -= 3;

if (CurrentFrameShield > 2)

CurrentFrameShield -= 2;

if (CurrentFrameDeath > 4)

{

CurrentFrameDeath -= 4;

death = false;

}

// Если обьект уничтожается то все последующие анимации в условии больше не проигрываются

if (destroy == false)

{

if (dx > 0)

{

// Воспроизведение анимации

sprite.setTextureRect(IntRect(96 \* int(CurrentFrameRun), 240, 96, 80));

left = 0;

}

if (ladderWalk == true)

{

sprite.setTextureRect(IntRect(96 \* int(CurrentFrameRun) + 96, 240, -96, 80));

}

if (dx < 0)

{

sprite.setTextureRect(IntRect(96 \* int(CurrentFrameRun) + 96, 240, -96, 80));

left = 1;

}

if ((dx == 0) and (attack == false)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameIdle) + 96, 160, -96, 80));

if ((dx == 0) and (left == 1) and (attack == false)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameIdle), 160, 96, 80));

if (attack == true) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameAttack), 0, 96, 80));

if ((attack == true) and (left == 1)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameAttack) + 96, 0, -96, 80));

if (animationDamage == true) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameDamage), 400, 96, 80));

if ((animationDamage == true) and (left == 1)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameDamage) + 96, 400, -96, 80));

if (shield == true) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameShield), 560, 96, 80));

if ((shield == true) and (left == 1)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameShield) + 96, 560, -96, 80));

}

if (death == true) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameDeath), 320, 96, 80));

if ((death == true) and (left == 1)) sprite.setTextureRect(IntRect(96 \* int(CurrentFrameDeath) + 96, 320, -96, 80));

// Позиция спрайта относительно центра

sprite.setPosition(rect.left - 58, rect.top - 98);

// Размер спрайта

sprite.setScale(2.5, 2.5);

dx = 0;

}

};

// Второй враг

class ENEMY2 {

public:

float dx, dy;

FloatRect rect;

bool onGround, enemyWalkLadderCollision = false, ladderWalk = false, attack = false, death = false, destroy = false, animationDamage = false, shield = false, level2 = false;

int left;

Sprite sprite;

float CurrentFrameRun, CurrentFrameAttack, CurrentFrameIdle, CurrentFrameDamage, CurrentFrameDeath, CurrentFrameShield, hp = 100;

ENEMY2(Texture& image)

{

sprite.setTexture(image);

rect = FloatRect(1280, 480, 44, 52);

dx = dy = 0;

CurrentFrameRun = 0;

CurrentFrameAttack = 0;

CurrentFrameIdle = 0;

CurrentFrameDamage = 0;

CurrentFrameDeath = 0;

CurrentFrameShield = 0;

}

////////////////////////////////////////////////////////////////////////

// Функция проверки столкновения с объектами в двумерном массиве TileMap

void Collision(int num)

{

for (int i = rect.top / sizeMapCollisionA; i < (rect.top + rect.height) / sizeMapCollisionA; i++)

for (int j = rect.left / sizeMapCollisionA; j < (rect.left + rect.width) / sizeMapCollisionA; j++)

{

// Проверка на размер карты массива

if ((j < 0) or (i < 0) or (j > 366) or (i > 128))

{

return;

}

if (level2 == false)

{

// Стена без свойств

if (TileMap[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

// Свойство шипов

if (TileMap[i][j] == 'H')

{

hp -= 1000;

}

}

if (level2 == true)

{

// Стена без свойств

if (TileMap2[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap2[i][j] == 'H')

{

hp -= 1000;

}

}

}

}

void update(float time)

{

float ground = 1768;

rect.left += dx \* time;

Collision(0);

if (!onGround) dy = dy + 0.0005 \* time;

rect.top += dy \* time;

onGround = false;

Collision(1);

if (rect.top > ground)

{

rect.top = ground;

dy = 0;

onGround = true;

};

CurrentFrameRun += 0.12;

CurrentFrameIdle += 0.12;

CurrentFrameAttack += 0.12;

CurrentFrameDamage += 0.13;

CurrentFrameDeath += 0.12;

if (CurrentFrameRun > 6)

CurrentFrameRun -= 6;

if (CurrentFrameIdle > 2)

CurrentFrameIdle -= 2;

if (CurrentFrameAttack > 6)

CurrentFrameAttack -= 6;

if (CurrentFrameDamage > 3)

CurrentFrameDamage -= 3;

if (CurrentFrameDeath > 4)

{

CurrentFrameDeath -= 4;

death = false;

}

if (destroy == false)

{

if (dx > 0)

{

sprite.setTextureRect(IntRect(160 \* int(CurrentFrameRun), 400, 160, 80));

left = 0;

}

if (ladderWalk == true)

{

sprite.setTextureRect(IntRect(160 \* int(CurrentFrameRun) + 160, 400, -160, 80));

}

if (dx < 0)

{

sprite.setTextureRect(IntRect(160 \* int(CurrentFrameRun) + 160, 400, -160, 80));

left = 1;

}

if ((dx == 0) and (attack == false)) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameIdle) + 160, 320, -160, 80));

if ((dx == 0) and (left == 1) and (attack == false)) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameIdle), 320, 160, 80));

if (attack == true) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameAttack), 0, 160, 80));

if ((attack == true) and (left == 1)) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameAttack) + 160, 0, -160, 80));

if (animationDamage == true) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameDamage), 240, 160, 80));

if ((animationDamage == true) and (left == 1)) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameDamage) + 160, 240, -160, 80));

}

if (death == true) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameDeath), 160, 160, 80));

if ((death == true) and (left == 1)) sprite.setTextureRect(IntRect(160 \* int(CurrentFrameDeath) + 160, 160, -160, 80));

sprite.setPosition(rect.left - 58, rect.top - 98);

sprite.setScale(2.5, 2.5);

dx = 0;

}

};

////////////////////////////////////////////////

// Первый персонаж с индивидуальными параметрами

class PLAYER {

public:

int level1();

float dx, dy, countDamage = 1, hud\_x, hud\_y;

FloatRect rect;

Texture hud\_r, hud\_hp, hud\_fon;

Sprite hud\_r\_sprite, hud\_hp\_sprite, hud\_fon\_sprite;

bool onGround, playerWalkLadderCollision = false, ladderWalk = false, animationDamage = false, jump, fall, run = false, verticalLadder = false, shield = false, gravity = true, death = false, attack = false, endGame = false, potion = true, potion2 = true, level2 = false, level3 = false, sound4 = false;

int left, position\_x = 50, position\_y = 580, i = 0;

Sprite sprite;

float currentFrameRun, currentFrameIdle, currentFrameJump, currentFrameShield, currentFrameDeath, currentFrameAttack, currentFrameFall, currentFrameDamage, currentFrameAttack2, hp = 100;

PLAYER(Texture& image)

{

sprite.setTexture(image);

rect = FloatRect(position\_x, position\_y, 104, 64); // (x (местоположение игрока), y (местоположение игрока), x, y)

dx = dy = 0;

currentFrameRun = 0;

currentFrameIdle = 0;

currentFrameJump = 4;

currentFrameDeath = 4;

currentFrameAttack = 4;

currentFrameDamage = 0;

currentFrameFall = 8;

currentFrameAttack2 = 9;

currentFrameShield = 0;

}

////////////////////////////////////////////////////////////////////////

// Функция проверки столкновения с объектами в двумерном массиве TileMap

void Collision(int num)

{

for (int i = rect.top / sizeMapCollisionA; i < (rect.top + rect.height) / sizeMapCollisionA; i++)

for (int j = rect.left / sizeMapCollisionA; j < (rect.left + rect.width) / sizeMapCollisionA; j++)

{

// Проверка на размер карты массива

if ((j < 0) or (i < 0) or (j > 366) or (i > 128))

{

return;

}

if ((level2 == false) and (level3 == false))

{

// Стена без свойств

if (TileMap[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

// Объект ступенька, толкает игрока наверх по dy в управлении

if (TileMap[i][j] == 'B')

{

if (dx > 0 && num == 0)

{

playerWalkLadderCollision = true;

ladderWalk = true;

}

else

{

playerWalkLadderCollision = false;

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if ((dx > 0) and (playerWalkLadderCollision = true))

{

dy = -0.188;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

else

{

ladderWalk = false;

}

// Первое зелье

if (TileMap[i][j] == 'E')

{

if ((hp < 100) and (potion == true))

{

sound4 = true;

potion = false;

hp = 100;

}

}

// Второе зелье

if (TileMap[i][j] == 'Q')

{

if ((hp < 100) and (potion2 == true))

{

sound4 = true;

potion2 = false;

hp = 100;

}

}

if (TileMap[i][j] == 'L')

{

if (dx > 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

playerWalkLadderCollision = true;

ladderWalk = true;

}

else

{

playerWalkLadderCollision = false;

ladderWalk = false;

}

if ((dx < 0) and (playerWalkLadderCollision = true))

{

dy = -0.188;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

if (TileMap[i][j] == 'I')

{

verticalLadder = true;

}

else

{

verticalLadder = false;

}

// Переход на другой уровень

if (TileMap[i][j] == 'S')

{

rect = FloatRect(1800, 1500, 104, 64);

level2 = true;

}

if (TileMap[i][j] == 'H')

{

hp -= 1000;

}

}

if ((level2 == true) and (level3 == false))

{

// Стена без свойств

if (TileMap2[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

// Объект ступенька, толкает игрока наверх по dy в управлении

if (TileMap2[i][j] == 'B')

{

if (dx > 0 && num == 0)

{

playerWalkLadderCollision = true;

ladderWalk = true;

}

else

{

playerWalkLadderCollision = false;

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if ((dx > 0) and (playerWalkLadderCollision = true))

{

dy = -0.188;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

else

{

ladderWalk = false;

}

if (TileMap2[i][j] == 'S')

{

rect = FloatRect(1800, 500, 104, 64);

level3 = true;

level2 = false;

}

if (TileMap2[i][j] == 'L')

{

if (dx > 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

playerWalkLadderCollision = true;

ladderWalk = true;

}

else

{

playerWalkLadderCollision = false;

ladderWalk = false;

}

if ((dx < 0) and (playerWalkLadderCollision = true))

{

dy = -0.188;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

// Вертикальная лестница

if (TileMap2[i][j] == 'I')

{

verticalLadder = true;

}

else

{

verticalLadder = false;

}

if (TileMap2[i][j] == 'H')

{

hp -= 1000;

}

}

if (level3 == true)

{

// Стена без свойств

if (TileMap3[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap3[i][j] == 'E')

{

if ((hp < 100) and (potion == true))

{

potion = false;

hp = 100;

}

}

if (TileMap3[i][j] == 'I')

{

verticalLadder = true;

}

else

{

verticalLadder = false;

}

}

}

}

void update(float time)

{

float ground = 2768;

hud\_x = rect.left += dx \* time;

Collision(0);

if (!onGround) dy = dy + 0.0008 \* time;

hud\_y = rect.top += dy \* time;

onGround = false;

Collision(1);

if (rect.top > ground)

{

rect.top = ground;

dy = 0;

onGround = true;

}

hud\_r.loadFromFile("sprites/hud/r.png");

hud\_hp.loadFromFile("sprites/hud/hp.png");

hud\_r\_sprite.setTexture(hud\_r);

hud\_r\_sprite.setPosition(hud\_x - 530, hud\_y - 380);

hud\_r\_sprite.setScale(180 / hud\_r.getSize().y, 180 / hud\_r.getSize().y);

hud\_hp\_sprite.setTexture(hud\_hp);

hud\_hp\_sprite.setPosition(hud\_x - 747 + hp \* 2.563, hud\_y - 346);

hud\_hp\_sprite.setScale(20 / hud\_hp.getSize().y, 20 / hud\_hp.getSize().y);

currentFrameRun += 0.136;

currentFrameIdle += 0.136;

currentFrameDeath += 0.136;

currentFrameShield += 0.136;

if (i == 0)

currentFrameAttack += 0.126;

currentFrameDamage += 0.126;

if (run == true)

currentFrameRun += 0.14;

if (i == 1)

currentFrameAttack2 += 0.126;

if (fall == true)

currentFrameFall += 0.166;

if (currentFrameRun > 8)

currentFrameRun -= 8;

if (currentFrameShield > 4)

currentFrameShield -= 3;

if (currentFrameFall > 14)

{

fall = false;

}

if (currentFrameIdle > 15)

currentFrameIdle -= 15;

if (currentFrameDeath > 15)

{

currentFrameDeath -= 11;

death = false;

}

if (currentFrameAttack > 9)

{

currentFrameAttack -= 9;

attack = false;

i = 1;

}

if (currentFrameAttack2 > 13)

{

currentFrameAttack2 -= 4;

shield = false;

i = 0;

}

if (endGame == false)

{

if (dx > 0)

{

sprite.setTextureRect(IntRect(128 \* int(currentFrameRun), 240, 128, 48));

left = 0;

}

if (ladderWalk == true)

{

sprite.setTextureRect(IntRect(128 \* int(currentFrameRun), 240, 128, 48));

}

if (dx < 0)

{

sprite.setTextureRect(IntRect(128 \* int(currentFrameRun) + 128, 240, -128, 48));

left = 1;

}

if ((dx == 0) and (fall == false)) sprite.setTextureRect(IntRect(128 \* int(currentFrameIdle), 96, 128, 48));

if ((dx == 0) and (left == 1) and (fall == false)) sprite.setTextureRect(IntRect(128 \* int(currentFrameIdle) + 128, 96, -128, 48));

if ((dy < 0) and (ladderWalk == false) and (jump == true)) sprite.setTextureRect(IntRect(128 \* int(currentFrameJump), 144, 128, 48));

if ((dy < 0) and (left == 1) and (ladderWalk == false) and (jump == true)) sprite.setTextureRect(IntRect(128 \* int(currentFrameJump) + 128, 144, -128, 48));

if ((dy > 0) and (ladderWalk == false)) sprite.setTextureRect(IntRect(128 \* int(8), 144, 128, 48));

if ((dy > 0) and (left == 1) and (ladderWalk == false)) sprite.setTextureRect(IntRect(128 \* int(8) + 128, 144, -128, 48));

if ((fall == true) and (ladderWalk == false)) sprite.setTextureRect(IntRect(128 \* int(currentFrameFall), 144, 128, 48));

if ((fall == true) and (left == 1) and (ladderWalk == false)) sprite.setTextureRect(IntRect(128 \* int(currentFrameFall) + 128, 144, -128, 48));

if ((attack == true) and (i == 0) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameAttack), 0, 128, 48));

if ((attack == true) and (i == 0) and (left == 1) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameAttack) + 128, 0, -128, 48));

if ((attack == true) and (i == 1) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameAttack2), 0, 128, 48));

if ((attack == true) and (i == 1) and (left == 1) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameAttack2) + 128, 0, -128, 48));

if ((shield == true) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameShield), 192, 128, 48));

if ((shield == true) and (left == 1) and (dx == 0) and (dy == 0)) sprite.setTextureRect(IntRect(128 \* int(currentFrameShield) + 128, 192, -128, 48));

}

if (death == true) sprite.setTextureRect(IntRect(128 \* int(currentFrameDeath), 48, 128, 48));

if ((death == true) and (left == 1)) sprite.setTextureRect(IntRect(128 \* int(currentFrameDeath) + 128, 48, -128, 48));

sprite.setPosition(rect.left - 120, rect.top - 80);

sprite.setScale(4, 4);

dx = 0;

}

};

class BOSS1 {

public:

float dx, dy;

FloatRect rect;

Sprite sprite;

bool onGround, enemyWalkLadderCollision = false, ladderWalk = false, attack = false, death = false, destroy = false, animationDamage = false, shield = false, level2 = false, level3 = false, boss1\_hud = false, staminaReload = false;

int left;

float CurrentFrameRun, CurrentFrameAttack, CurrentFrameIdle, CurrentFrameDamage, CurrentFrameDeath, CurrentFrameShield, hp = 200, stamina = 50;

BOSS1(Texture& image)

{

sprite.setTexture(image);

rect = FloatRect(1280, 1480, 44, 52);

dx = dy = 0;

CurrentFrameRun = 0;

CurrentFrameAttack = 0;

CurrentFrameIdle = 0;

CurrentFrameDamage = 0;

CurrentFrameDeath = 0;

CurrentFrameShield = 0;

}

////////////////////////////////////////////////////////////////////////

// Функция проверки столкновения с объектами в двумерном массиве TileMap

void Collision(int num)

{

for (int i = rect.top / sizeMapCollisionA; i < (rect.top + rect.height) / sizeMapCollisionA; i++)

for (int j = rect.left / sizeMapCollisionA; j < (rect.left + rect.width) / sizeMapCollisionA; j++)

{

// Проверка на размер карты массива

if ((j < 0) or (i < 0) or (j > 366) or (i > 128))

{

return;

}

if ((level2 == false) and (level3 == false))

{

// Стена без свойств

if (TileMap[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap[i][j] == 'J')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap[i][j] == 'B')

{

if (dx > 0 && num == 0)

{

enemyWalkLadderCollision = true;

ladderWalk = true;

}

else

{

enemyWalkLadderCollision = false;

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if ((dx > 0) and (enemyWalkLadderCollision = true))

{

dy = -0.14;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

if (TileMap[i][j] == 'L')

{

if (dx > 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

enemyWalkLadderCollision = true;

ladderWalk = true;

}

else

{

enemyWalkLadderCollision = false;

ladderWalk = false;

}

if ((dx < 0) and (enemyWalkLadderCollision = true))

{

dy = -0.14;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

if (TileMap[i][j] == 'H')

{

hp -= 1000;

}

}

if ((level2 == true) and (level3 == false))

{

// Стена без свойств

if (TileMap2[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap2[i][j] == 'J')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

if (TileMap2[i][j] == 'B')

{

if (dx > 0 && num == 0)

{

enemyWalkLadderCollision = true;

ladderWalk = true;

}

else

{

enemyWalkLadderCollision = false;

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if ((dx > 0) and (enemyWalkLadderCollision = true))

{

dy = -0.14;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

if (TileMap2[i][j] == 'L')

{

if (dx > 0 && num == 0)

{

ladderWalk = true;

}

else

{

ladderWalk = false;

}

if (dx < 0 && num == 0)

{

enemyWalkLadderCollision = true;

ladderWalk = true;

}

else

{

enemyWalkLadderCollision = false;

ladderWalk = false;

}

if ((dx < 0) and (enemyWalkLadderCollision = true))

{

dy = -0.14;

ladderWalk = true;

}

else

{

ladderWalk = false;

}

}

if (TileMap2[i][j] == 'H')

{

hp -= 1000;

}

}

if ((level2 == false) and (level3 == true))

{

// Стена без свойств

if (TileMap3[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

}

}

}

void update(float time)

{

float ground = 1768;

rect.left += dx \* time;

Collision(0);

if (!onGround) dy = dy + 0.0005 \* time;

rect.top += dy \* time;

onGround = false;

Collision(1);

if (rect.top > ground)

{

rect.top = ground;

dy = 0;

onGround = true;

};

CurrentFrameRun += 0.12;

CurrentFrameIdle += 0.12;

CurrentFrameAttack += 0.12;

CurrentFrameDamage += 0.13;

CurrentFrameDeath += 0.12;

if (CurrentFrameRun > 12)

CurrentFrameRun -= 12;

if (CurrentFrameIdle > 4)

CurrentFrameIdle -= 4;

if (CurrentFrameAttack > 8)

CurrentFrameAttack -= 8;

if (CurrentFrameDamage > 3)

CurrentFrameDamage -= 3;

if (CurrentFrameDeath > 13)

{

CurrentFrameDeath -= 13;

death = false;

}

if (destroy == false)

{

if (dx > 0)

{

sprite.setTextureRect(IntRect(64 \* int(CurrentFrameRun), 129, 64, 48));

left = 0;

}

if (dx < 0)

{

sprite.setTextureRect(IntRect(64 \* int(CurrentFrameRun) + 64, 129, -64, 48));

left = 1;

}

if ((dx == 0) and (attack == false)) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameIdle) + 64, 0, -64, 48));

if ((dx == 0) and (left == 1) and (attack == false)) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameIdle), 0, 64, 48));

if (attack == true) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameAttack), 0, 64, 48));

if ((attack == true) and (left == 1)) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameAttack) + 64, 0, -64, 48));

if (animationDamage == true) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameDamage), 257, 64, 48));

if ((animationDamage == true) and (left == 1)) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameDamage) + 64, 257, -64, 48));

}

if (death == true) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameDeath), 65, 64, 48));

if ((death == true) and (left == 1)) sprite.setTextureRect(IntRect(64 \* int(CurrentFrameDeath) + 64, 65, -64, 48));

sprite.setPosition(rect.left - 56, rect.top - 42);

sprite.setScale(4.5, 4.5);

dx = 0;

}

};

class BOSS2 {

public:

float dx, dy;

FloatRect rect;

Sprite sprite;

bool onGround, enemyWalkLadderCollision = false, ladderWalk = false, attack = false, death = false, destroy = false, animationDamage = false, shield = false, level2 = false, level3 = false, boss1\_hud = false, staminaReload = false;

int left;

float CurrentFrameRun, CurrentFrameAttack, CurrentFrameIdle, CurrentFrameDamage, CurrentFrameDeath, CurrentFrameShield, hp = 200, stamina = 50;

BOSS2(Texture& image)

{

sprite.setTexture(image);

rect = FloatRect(1280, 1480, 44, 52);

dx = dy = 0;

CurrentFrameRun = 0;

CurrentFrameAttack = 0;

CurrentFrameIdle = 0;

CurrentFrameDamage = 0;

CurrentFrameDeath = 0;

CurrentFrameShield = 0;

}

////////////////////////////////////////////////////////////////////////

// Функция проверки столкновения с объектами в двумерном массиве TileMap

void Collision(int num)

{

for (int i = rect.top / sizeMapCollisionA; i < (rect.top + rect.height) / sizeMapCollisionA; i++)

for (int j = rect.left / sizeMapCollisionA; j < (rect.left + rect.width) / sizeMapCollisionA; j++)

{

// Проверка на размер карты массива

if ((j < 0) or (i < 0) or (j > 366) or (i > 128))

{

return;

}

// Стена без свойств

if (TileMap3[i][j] == 'A')

{

if (dy > 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA - rect.height;

dy = 0;

onGround = true;

}

if (dy < 0 && num == 1)

{

rect.top = i \* sizeMapCollisionA + sizeMapCollisionA;

dy = 0;

}

if (dx > 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA - rect.width;

}

if (dx < 0 && num == 0)

{

rect.left = j \* sizeMapCollisionA + sizeMapCollisionA;

}

}

}

}

void update(float time)

{

float ground = 1768;

rect.left += dx \* time;

Collision(0);

if (!onGround) dy = dy + 0.0005 \* time;

rect.top += dy \* time;

onGround = false;

Collision(1);

if (rect.top > ground)

{

rect.top = ground;

dy = 0;

onGround = true;

};

CurrentFrameRun += 0.12;

CurrentFrameIdle += 0.12;

CurrentFrameAttack += 0.12;

CurrentFrameDamage += 0.13;

CurrentFrameDeath += 0.12;

if (CurrentFrameRun > 8)

CurrentFrameRun -= 8;

if (CurrentFrameIdle > 9)

CurrentFrameIdle -= 9;

if (CurrentFrameAttack > 7)

CurrentFrameAttack -= 7;

if (CurrentFrameDamage > 3)

CurrentFrameDamage -= 3;

if (CurrentFrameDeath > 7)

{

CurrentFrameDeath -= 7;

death = false;

}

cout << "CADRS: " << CurrentFrameIdle << endl;

if (destroy == false)

{

if (dx > 0)

{

sprite.setTextureRect(IntRect(144 \* int(CurrentFrameRun), 576, 144, 96));

left = 0;

}

if (dx < 0)

{

sprite.setTextureRect(IntRect(144 \* int(CurrentFrameRun) + 144, 576, -144, 96));

left = 1;

}

if ((dx == 0) and (attack == false)) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameIdle), 384, 144, 96));

if ((dx == 0) and (left == 1) and (attack == false)) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameIdle) + 144, 384, -144, 96));

if (attack == true) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameAttack), 96, 144, 96));

if ((attack == true) and (left == 1)) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameAttack) + 144, 96, -144, 96));

if (animationDamage == true) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameDamage), 480, 144, 96));

if ((animationDamage == true) and (left == 1)) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameDamage) + 144, 480, -144, 96));

}

if (death == true) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameDeath), 288, 144, 96));

if ((death == true) and (left == 1)) sprite.setTextureRect(IntRect(144 \* int(CurrentFrameDeath) + 144, 288, -144, 96));

sprite.setPosition(rect.left - 300, rect.top - 335);

sprite.setScale(4.5, 4.5);

dx = 0;

}

};

void keyboard()

{

if (GetAsyncKeyState(27)) //esc выход

exit(0);

}

void menu(RenderWindow& window)

{

Texture menuTexture1, menuTexture2, menuTexture3, aboutTexture, menuBackground;

menuTexture1.loadFromFile("sprites/buttons/start.png");

menuTexture2.loadFromFile("sprites/buttons/load.png");

menuTexture3.loadFromFile("sprites/buttons/exit.png");

aboutTexture.loadFromFile("sprites/backgrounds/gothic\_background/background.png");

menuBackground.loadFromFile("sprites/backgrounds/gothic\_background/background.png");

Sprite menu1(menuTexture1), menu2(menuTexture2), menu3(menuTexture3), about(aboutTexture), menuBg(menuBackground);

bool isMenu = 1, choiseMenu = 1;

int menuNum = 0, choiseNum = 0;

menu1.setPosition(560, 50);

menu1.setScale(2, 2);

menu2.setPosition(555, 110);

menu2.setScale(2, 2);

menu3.setPosition(560, 170);

menu3.setScale(2, 2);

menuBg.setPosition(0, 0);

menuBg.setScale(3.4, 3.4);

// MUSIC

Music music;

music.openFromFile("music/menu.ogg");

music.play();

music.setLoop(true);

music.setVolume(25);

// MENU

while (isMenu)

{

Event event;

while (window.pollEvent(event))

{

if (event.type == Event::Closed)

window.close();

else if (event.type == Event::Resized)

{

cout << "resize: (" << event.size.width << ',' << event.size.height << ") -> " << endl;

}

}

menu1.setColor(Color::White);

menu2.setColor(Color::White);

menu3.setColor(Color::White);

menuNum = 0;

window.clear(Color(129, 181, 221));

if (IntRect(560, 50, 300, 50).contains(Mouse::getPosition(window)))

{

menu1.setColor(Color::Blue);

menuNum = 1;

}

if (IntRect(555, 110, 300, 50).contains(Mouse::getPosition(window)))

{

menu2.setColor(Color::Blue);

menuNum = 2;

}

if (IntRect(560, 170, 300, 50).contains(Mouse::getPosition(window)))

{

menu3.setColor(Color::Blue);

menuNum = 3;

}

if (Mouse::isButtonPressed(Mouse::Left))

{

if (menuNum == 1)

{

isMenu = false;

}

if (menuNum == 2)

{

window.draw(about);

window.display();

while (!Keyboard::isKeyPressed(Keyboard::Escape));

}

if (menuNum == 3)

{

window.close(); isMenu = false;

}

}

if ((IntRect(560, 50, 300, 50).contains(Mouse::getPosition(window))) and (Mouse::isButtonPressed(Mouse::Left)))

{

isMenu = false;

}

window.draw(menuBg);

window.draw(menu1);

window.draw(menu2);

window.draw(menu3);

window.display();

}

}

float timer;

// В x и y центральные точки объектов

// view\_radius — радиус взгляда

bool is\_player\_in\_view(int player\_x, int player\_y, int enemy\_x, int enemy\_y, int view\_radius)

{

int x = player\_x - enemy\_x;

int y = player\_y - enemy\_y;

return x \* x + y \* y < view\_radius\* view\_radius;

}

void main()

{

map1();

map2();

map3();

ContextSettings settings;

settings.antialiasingLevel = 8;

RenderWindow window(VideoMode(1280, 720), "Bastion of mind", Style::None, settings); // Style::Default

window.setFramerateLimit(60);

menu(window); //вызов меню

Event event;

View view;

Texture t, t2, enemy1, enemy2, level\_collision, level\_background, level\_background\_fon, hp, hp2, level\_collision2, level\_background2, level\_collision3, level\_background3, boss1, boss2;

Sprite level\_collision\_s, level\_background\_s, level\_background\_fon\_sprite, potion\_hp, potion\_hp2, level\_collision2\_s, level\_background2\_s, level\_collision3\_s, level\_background3\_s;

level\_collision.loadFromFile("sprites/levels/level\_1.png");

level\_background.loadFromFile("sprites/levels/level\_1\_fon.png");

level\_collision2.loadFromFile("sprites/levels/level\_2.png");

level\_background2.loadFromFile("sprites/levels/level\_2\_fon.png");

level\_collision3.loadFromFile("sprites/levels/loka\_3\_plat.png");

level\_background3.loadFromFile("sprites/levels/loka\_3\_fon.png");

level\_background\_fon.loadFromFile("sprites/levels/background\_level\_1.png");

hp.loadFromFile("sprites/entities/potions/redPotion.png");

hp2.loadFromFile("sprites/entities/potions/redPotion.png");

// PLATFORM

level\_collision\_s.setTexture(level\_collision);

level\_collision\_s.setPosition(15, -300); // 0, 31

level\_collision\_s.setScale(3000 / level\_collision.getSize().y, 3000 / level\_collision.getSize().y);

level\_collision2\_s.setTexture(level\_collision2);

level\_collision2\_s.setPosition(15, -980); // 0, 31

level\_collision2\_s.setScale(5000 / level\_collision2.getSize().y, 5000 / level\_collision2.getSize().y);

level\_collision3\_s.setTexture(level\_collision3);

level\_collision3\_s.setPosition(15, -218); // 0, 31

level\_collision3\_s.setScale(3000 / level\_collision3.getSize().y, 3000 / level\_collision3.getSize().y);

// BACKGROUND

level\_background\_s.setTexture(level\_background);

level\_background\_s.setPosition(15, -300);

level\_background\_s.setScale(3000 / level\_background.getSize().y, 3000 / level\_background.getSize().y);

level\_background2\_s.setTexture(level\_background2);

level\_background2\_s.setPosition(15, -980);

level\_background2\_s.setScale(5000 / level\_background2.getSize().y, 5000 / level\_background2.getSize().y);

level\_background\_fon\_sprite.setTexture(level\_background\_fon);

level\_background\_fon\_sprite.setPosition(-800, -800);

level\_background\_fon\_sprite.setScale(2000 / level\_background\_fon.getSize().y, 2000 / level\_background\_fon.getSize().y);

level\_background3\_s.setTexture(level\_background3);

level\_background3\_s.setPosition(15, -218);

level\_background3\_s.setScale(3000 / level\_background3.getSize().y, 3000 / level\_background3.getSize().y);

// ITEMS

potion\_hp.setTexture(hp);

potion\_hp.setPosition(2690, 710);

potion\_hp.setScale(80 / hp.getSize().y, 80 / hp.getSize().y);

potion\_hp2.setTexture(hp2);

potion\_hp2.setPosition(1300, 1898);

potion\_hp2.setScale(80 / hp2.getSize().y, 80 / hp2.getSize().y);

// CHARACTERS

t.loadFromFile("sprites/entities/characters/knights/Knight\_1.png");

t2.loadFromFile("sprites/entities/characters/knights/knight\_female/Sprites/knight.png");

enemy1.loadFromFile("sprites/entities/characters/skeletons/skeloton1.png");

enemy2.loadFromFile("sprites/entities/characters/skeletons/skeleton\_3.png");

boss1.loadFromFile("sprites/entities/characters/skeletons/Skeleton\_enemy.png");

boss2.loadFromFile("sprites/entities/characters/skeletons/boss\_2.png");

PLAYER p(t);

ENEMY e(enemy1);

ENEMY2 e2(enemy2);

BOSS1 b1(boss1);

BOSS2 b2(boss2);

Clock clock;

RectangleShape rectangle;

rectangle.setSize(sf::Vector2f(32, 32));

// PLAYER HIT

SoundBuffer buffer;

buffer.loadFromFile("sounds/Ydar\_mehem.ogg");

Sound sound;

sound.setBuffer(buffer);

sound.setVolume(25);

bool soundWalk = false;

// WALK

SoundBuffer hodba\_popoly;

hodba\_popoly.loadFromFile("sounds/po\_betonu.ogg");

Sound sound2;

sound2.setBuffer(hodba\_popoly);

sound2.setVolume(40);

// SKELETON HIT

SoundBuffer ydar\_skeleta;

ydar\_skeleta.loadFromFile("sounds/ydar\_skeleta.ogg");

Sound sound3;

sound3.setBuffer(ydar\_skeleta);

sound3.setVolume(25);

// DRINK VODKA

SoundBuffer drink\_vodka;

drink\_vodka.loadFromFile("sounds/drink\_vodka.ogg");

Sound sound4;

sound4.setBuffer(drink\_vodka);

sound4.setVolume(25);

// MUSIC

Music music;

music.openFromFile("music/music\_level\_1.ogg");

music.play();

music.setLoop(true);

music.setVolume(25);

cout << "main start" << endl;

this\_thread::sleep\_for(chrono::milliseconds(3000));

cout << this\_thread::get\_id() << endl;

cout << "main end" << endl;

Texture hud\_r, hud\_hp;

Sprite hud\_r\_sprite, hud\_hp\_sprite;

hud\_r.loadFromFile("sprites/hud/boss\_r.png");

hud\_hp.loadFromFile("sprites/hud/boss\_hp.png");

int hight = 0;

while (window.isOpen())

{

keyboard();

hud\_r\_sprite.setTexture(hud\_r);

hud\_r\_sprite.setPosition(p.hud\_x - 550, p.hud\_y + 320);

hud\_r\_sprite.setScale(80 / hud\_r.getSize().y, 80 / hud\_r.getSize().y);

hud\_hp\_sprite.setTexture(hud\_hp);

hud\_hp\_sprite.setPosition(p.hud\_x - 952 + b1.hp \* 2.35, p.hud\_y + 345);

hud\_hp\_sprite.setScale(20 / hud\_hp.getSize().y, 20 / hud\_hp.getSize().y);

Vector2f player\_position(p.position\_x, p.position\_y);

player\_position += Vector2f(10, 10);

ofstream ofs{ "positions.sav" };

ofs << player\_position.x << ' ' << player\_position.y << endl;

while (window.pollEvent(event))

{

if (event.type == Event::Closed)

window.close();

if (event.type == Event::Resized)

window.create(VideoMode(1280, 720), "Bastion of mind", Style::Fullscreen, settings);

}

if (p.endGame == false)

{

if (Keyboard::isKeyPressed(Keyboard::D))

{

p.dx = 0.15;

p.run = false;

}

if (Keyboard::isKeyPressed(Keyboard::A))

{

p.dx = -0.15;

p.run = false;

}

if (Keyboard::isKeyPressed(Keyboard::Space))

{

if (p.onGround)

{

p.dy = -0.5;

p.onGround = false;

p.jump = true;

p.fall = false;

p.currentFrameFall -= 5;

}

}

if ((p.onGround == true) and (p.currentFrameFall <= 14))

p.fall = true;

if (p.dx != 0)

{

// soundWalk = false;

}

if (p.verticalLadder == true)

{

if (Keyboard::isKeyPressed(Keyboard::W))

{

p.dy = -0.15;

p.run = false;

// p.onGround = true;

}

if (Keyboard::isKeyPressed(Keyboard::S))

{

p.dy = 0.15;

p.run = false;

// p.onGround = true;

}

if (!Keyboard::isKeyPressed(Keyboard::W) && !Keyboard::isKeyPressed(Keyboard::S))

{

p.dy = 0;

}

}

if (Keyboard::isKeyPressed(Keyboard::LShift) && Keyboard::isKeyPressed(Keyboard::D) && p.onGround == true)

{

p.dx = 0.5;

p.run = true;

}

if (Keyboard::isKeyPressed(Keyboard::LShift) && Keyboard::isKeyPressed(Keyboard::A) && p.onGround == true)

{

p.dx = -0.5;

p.run = true;

}

if (!Keyboard::isKeyPressed(Keyboard::D) && !Keyboard::isKeyPressed(Keyboard::A) && p.onGround == true)

{

sound2.play();

}

if (Keyboard::isKeyPressed(Keyboard::F) && p.onGround == true && p.attack == false && p.shield == false)

{

p.attack = true;

sound.play();

}

if (Keyboard::isKeyPressed(Keyboard::LControl) && p.onGround == true)

{

p.shield = true;

}

}

if (e.onGround == false)

{

e.dy = 0.4;

}

if ((e.animationDamage == true) or (e2.animationDamage == true))

{

sound3.play();

}

int pcenterx;

int pcentery;

int ecenterx = e.rect.left + e.rect.width / 2;

int ecentery = e.rect.top + e.rect.height / 2;

int ecenterx2 = e2.rect.left + e2.rect.width / 2;

int ecentery2 = e2.rect.top + e2.rect.height / 2;

int ecenterx3 = b1.rect.left + b1.rect.width / 2;

int ecentery3 = b1.rect.top + b1.rect.height / 2;

int ecenterx4 = b2.rect.left + b2.rect.width / 2;

int ecentery4 = b2.rect.top + b2.rect.height / 2;

// Область зрения

int view\_radius = 400;

pcenterx = p.rect.left + p.rect.width / 2 + 65;

pcentery = p.rect.top + p.rect.height / 2;

if (p.level2 == false)

{

if ((is\_player\_in\_view(pcenterx, pcentery, ecenterx, ecentery, view\_radius)) and (e.destroy == false))

{

if ((ecenterx < pcenterx) and (p.animationDamage == false))

{

e.dx = 0.15;

}

if ((ecenterx > pcenterx) and (p.animationDamage == false))

{

e.dx = -0.15;

}

int distance = abs(ecenterx - pcenterx);

///////////////////////////////////////

// Отношения врага к игроку ))

if (distance < 80)

{

e.dx = 0;

if (p.hp >= 0)

e.attack = true;

else

e.attack = false;

if (((e.CurrentFrameAttack > 11) or (e.CurrentFrameAttack > 5)) and (p.hp > 0) and (e.animationDamage == false))

{

p.hp -= 0.20;

p.animationDamage = true;

cout << p.hp << endl;

}

else

{

p.animationDamage = false;

}

if (((p.currentFrameAttack > 6) or (p.currentFrameAttack2 > 11)) and (p.attack == true) and (p.animationDamage == false))

{

e.hp -= p.countDamage;

e.animationDamage = true;

cout << e.hp << endl;

}

else

{

e.animationDamage = false;

}

}

else

{

e.attack = false;

p.animationDamage = false;

e.animationDamage = false;

}

}

if ((is\_player\_in\_view(pcenterx, pcentery, ecenterx2, ecentery2, view\_radius)) and (e2.destroy == false))

{

if ((ecenterx2 < pcenterx) and (p.animationDamage == false))

{

e2.dx = 0.15;

}

if ((ecenterx2 > pcenterx) and (p.animationDamage == false))

{

e2.dx = -0.15;

}

int distance = abs(ecenterx2 - pcenterx);

///////////////////////////////////////

// Отношения врага к игроку ))

if (distance < 80)

{

e2.dx = 0;

if (p.hp >= 0)

e2.attack = true;

else

e2.attack = false;

if (((e2.CurrentFrameAttack > 11) or (e2.CurrentFrameAttack > 5)) and (p.hp > 0) and (e2.animationDamage == false))

{

p.hp -= 2;

p.animationDamage = true;

cout << p.hp << endl;

}

else

{

p.animationDamage = false;

}

if (((p.currentFrameAttack > 6) or (p.currentFrameAttack2 > 11)) and (p.attack == true) and (p.animationDamage == false))

{

e2.hp -= p.countDamage;

e2.animationDamage = true;

cout << e2.hp << endl;

}

else

{

e2.animationDamage = false;

}

}

else

{

e2.attack = false;

p.animationDamage = false;

e2.animationDamage = false;

}

}

if ((is\_player\_in\_view(pcenterx, pcentery, ecenterx3, ecentery3, view\_radius)) and (b1.destroy == false))

{

if ((ecenterx3 < pcenterx) and (p.animationDamage == false))

{

b1.dx = 0.15;

}

if ((ecenterx3 > pcenterx) and (p.animationDamage == false))

{

b1.dx = -0.15;

}

int distance = abs(ecenterx3 - pcenterx);

///////////////////////////////////////

// Отношения врага к игроку ))

if (distance < 80)

{

b1.dx = 0;

if ((p.hp >= 0) and (b1.stamina > 0))

b1.attack = true;

else

b1.attack = false;

if (((b1.CurrentFrameAttack > 11) or (b1.CurrentFrameAttack > 5)) and (p.hp > 0) and (b1.animationDamage == false) and (p.shield == false) and (b1.stamina > 0))

{

p.hp -= 2;

p.animationDamage = true;

cout << p.hp << endl;

b1.stamina -= 25;

}

else

{

p.animationDamage = false;

}

cout << "Stamina: " << b1.stamina << endl;

if (b1.stamina >= 100)

b1.staminaReload = false;

if (b1.stamina <= 0)

b1.stamina += 0.2;

if ((b1.stamina <= 0) and (b1.staminaReload == false))

{

b1.stamina -= 100;

b1.staminaReload = true;

}

if (((p.currentFrameAttack > 6) or (p.currentFrameAttack2 > 11)) and (p.attack == true) and (p.animationDamage == false))

{

b1.hp -= p.countDamage;

b1.animationDamage = true;

cout << b1.hp << endl;

}

else

{

b1.animationDamage = false;

}

}

else

{

b1.attack = false;

p.animationDamage = false;

b1.animationDamage = false;

}

if (distance < 400)

b1.boss1\_hud = true;

if (distance >= 400)

b1.boss1\_hud = false;

}

if ((is\_player\_in\_view(pcenterx, pcentery, ecenterx4, ecentery4, view\_radius)) and (b2.destroy == false))

{

if ((ecenterx4 < pcenterx) and (p.animationDamage == false))

{

b2.dx = 0.25;

}

if ((ecenterx4 > pcenterx) and (p.animationDamage == false))

{

b2.dx = -0.25;

}

int distance = abs(ecenterx4 - pcenterx);

///////////////////////////////////////

// Отношения врага к игроку ))

if (distance < 120)

{

b2.dx = 0;

if ((p.hp >= 0) and (b2.stamina > 0))

b2.attack = true;

else

b2.attack = false;

if ((b2.CurrentFrameAttack > 2) and (p.hp > 0) and (b2.animationDamage == false) and (p.shield == false) and (b2.stamina > 0))

{

p.hp -= 15;

p.animationDamage = true;

cout << p.hp << endl;

b2.stamina -= 15;

}

else

{

p.animationDamage = false;

}

cout << "Stamina: " << b2.stamina << endl;

if (b2.stamina >= 100)

b2.staminaReload = false;

if (b2.stamina <= 0)

b2.stamina += 0.2;

if ((b2.stamina <= 0) and (b2.staminaReload == false))

{

b2.stamina -= 100;

b2.staminaReload = true;

}

if (((p.currentFrameAttack > 6) or (p.currentFrameAttack2 > 11)) and (p.attack == true) and (p.animationDamage == false))

{

b2.hp -= p.countDamage;

b2.animationDamage = true;

cout << b2.hp << endl;

}

else

{

b2.animationDamage = false;

}

}

else

{

b2.attack = false;

p.animationDamage = false;

b2.animationDamage = false;

}

if (distance < 400)

b2.boss1\_hud = true;

if (distance >= 400)

b2.boss1\_hud = false;

}

}

if ((p.endGame == false) and (p.hp <= 0))

{

p.death = true;

p.endGame = true;

}

if ((e.destroy == false) and (e.hp <= 0))

{

e.death = true;

e.destroy = true;

}

if ((e2.destroy == false) and (e2.hp <= 0))

{

e2.death = true;

e2.destroy = true;

}

if ((b1.destroy == false) and (b1.hp <= 0))

{

b1.death = true;

b1.destroy = true;

}

if ((b2.destroy == false) and (b2.hp <= 0))

{

b2.death = true;

b2.destroy = true;

}

float time = clock.getElapsedTime().asMilliseconds();

clock.restart();

timer += time;

float delay = 0.005;

if (timer > delay)

{

p.update(timer);

e.update(timer);

e2.update(timer);

b1.update(timer);

b2.update(timer);

timer = 0;

}

view.setCenter(p.sprite.getPosition().x + 60 \* p.sprite.getScale().x, p.sprite.getPosition().y + 24 \* p.sprite.getScale().y);

view.setSize(1280, 768);

window.setView(view);

window.clear(Color::White);

if (p.sound4 == true)

{

sound4.play();

p.sound4 = false;

}

if (p.level2 == false)

{

window.draw(level\_background\_fon\_sprite);

window.draw(level\_background\_s);

window.draw(level\_collision\_s);

window.draw(e.sprite);

window.draw(e2.sprite);

window.draw(b1.sprite);

if (p.potion == true)

{

window.draw(potion\_hp);

}

if (p.potion2 == true)

{

window.draw(potion\_hp2);

}

}

else

{

window.draw(level\_background\_fon\_sprite);

window.draw(level\_background2\_s);

window.draw(level\_collision2\_s);

}

if (p.level3 == true)

{

window.draw(level\_background3\_s);

window.draw(level\_collision3\_s);

b1.level3 = true;

b1.level2 = false;

window.draw(b1.sprite);

window.draw(b2.sprite);

if (hight == 0)

{

b1.rect = FloatRect(1280, 1480, 44, 126);

hight = 1;

}

}

window.draw(p.sprite);

window.draw(p.hud\_hp\_sprite);

window.draw(p.hud\_r\_sprite);

if (b1.boss1\_hud == true)

{

window.draw(hud\_hp\_sprite);

window.draw(hud\_r\_sprite);

}

window.display();

}

}

# Практическое занятие №19. Работа с сетевым подключением, сокетами, и многопоточностью.

**Листинг 70.**

Выполнил: Пихтов Виталий Алексеевич

Программа консольный чат типа Клиент-Сервер. Сервер слушает передачу пакетов, и передает пакеты от одного пользователя другим. Использовались библиотеки WINSOCK2, SDKDDKVER, STRING.

#include <SDKDDKVer.h>

#pragma comment(lib, "ws2\_32.lib")

#include <winsock2.h>

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <iostream>

#include <string>

#pragma warning(disable: 4996)

SOCKET Connections[100];

int Counter = 0;

enum Packet

{

Chat,

P\_Test

};

bool ProcessPacket(int index, Packet packettype)

{

switch (packettype)

{

case Chat:

{

int msg\_size;

recv(Connections[index], (char\*)&msg\_size, sizeof(int), NULL);

char\* msg = new char[msg\_size + 1];

msg[msg\_size] = '\0';

recv(Connections[index], msg, msg\_size, NULL);

for (int i = 0; i < Counter; i++)

{

if (i == index)

{

continue;

}

Packet msgtype = Chat;

send(Connections[i], (char\*)&msgtype, sizeof(Packet), NULL);

send(Connections[i], (char\*)&msg\_size, sizeof(int), NULL);

send(Connections[i], msg, msg\_size, NULL);

}

delete[] msg;

break;

}

case P\_Test:

{

int msg\_size;

recv(Connections[index], (char\*)&msg\_size, sizeof(int), NULL);

char\* msg = new char[msg\_size + 1];

msg[msg\_size] = '\0';

recv(Connections[index], msg, msg\_size, NULL);

for (int i = 0; i < Counter; i++)

{

if (i == index)

{

continue;

}

Packet msgtype = P\_Test;

send(Connections[i], (char\*)&msgtype, sizeof(Packet), NULL);

send(Connections[i], (char\*)&msg\_size, sizeof(int), NULL);

send(Connections[i], msg, msg\_size, NULL);

}

delete[] msg;

break;

}

default:

std::cout << "Unrecognized packet: " << packettype << std::endl;

break;

}

return true;

}

void ClientHandler(int index)

{

Packet packettype;

while (true)

{

recv(Connections[index], (char\*)&packettype, sizeof(Packet), NULL);

if (!ProcessPacket(index, packettype))

{

break;

}

}

closesocket(Connections[index]);

}

int main(int argc, char\* argv[])

{

WSAData wsaData;

WORD DLLVersion = MAKEWORD(2, 1);

if (WSAStartup(DLLVersion, &wsaData) != 0)

{

std::cout << "Error >~<" << std::endl;

exit(1);

}

char ip[256] = "0.0.0.0";

int port;

std::cout << "ENTER PORT: " << std::endl;

std::cin >> port;

SOCKADDR\_IN addr;

int sizeofaddr = sizeof(addr);

addr.sin\_addr.s\_addr = inet\_addr(ip);

addr.sin\_port = htons(port);

addr.sin\_family = AF\_INET;

SOCKET sListen = socket(AF\_INET, SOCK\_STREAM, NULL);

bind(sListen, (SOCKADDR\*)&addr, sizeof(addr));

listen(sListen, SOMAXCONN);

SOCKET newConnection;

for (int i = 0; i < 100; i++)

{

newConnection = accept(sListen, (SOCKADDR\*)&addr, &sizeofaddr);

if (newConnection == 0)

{

std::cout << "Error #2 >~<\n";

}

else {

std::cout << "Client Connected !\n";

std::string msg = "Hello OwO";

int msg\_size = msg.size();

Packet msgtype = Chat;

send(newConnection, (char\*)&msgtype, sizeof(Packet), NULL);

send(newConnection, (char\*)&msg\_size, sizeof(int), NULL);

send(newConnection, msg.c\_str(), msg\_size, NULL);

Connections[i] = newConnection;

Counter++;

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)ClientHandler, (LPVOID)(i), NULL, NULL);

Packet testpacket = P\_Test;

send(newConnection, (char\*)&testpacket, sizeof(Packet), NULL);

}

}

system("pause");

return 0;

}

**Листинг 71.**

Выполнил: Пихтов Виталий Алексеевич

Программа консольный чат типа Клиент-Сервер. Клиент слушает передачу пакетов от сервера других подключенных клиентов, и передает пакеты на сервер. Использовались библиотеки WINSOCK2, SDKDDKVER, STRING, MATH, FSTREAM.

#define NOMINMAX

#include <SDKDDKVer.h>

#pragma comment(lib, "ws2\_32.lib")

#include <winsock2.h>

#include <iostream>

#include <string>

#include <math.h>

#include <fstream>

#include <string>

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#pragma warning(disable: 4996)

SOCKET Connection;

std::string message = " ";

char mode = ' ';

enum Packet

{

Chat,

P\_Test,

};

bool ProcessPacket(Packet packettype)

{

switch (packettype)

{

case Chat:

{

int msg\_size;

recv(Connection, (char\*)&msg\_size, sizeof(int), NULL);

char\* msg = new char[msg\_size + 1];

msg[msg\_size] = '\0';

recv(Connection, msg, msg\_size, NULL);

message = msg;

std::cout << msg << std::endl;

delete[] msg;

break;

}

case P\_Test:

std::cout << "Test packet :3.\n";

break;

default:

std::cout << "Unrecognized packet >~<: " << packettype << std::endl;

break;

}

return true;

}

void ClientHandler()

{

Packet packettype;

while (true)

{

recv(Connection, (char\*)&packettype, sizeof(Packet), NULL);

if (!ProcessPacket(packettype))

{

break;

}

}

closesocket(Connection);

}

float timer;

int main(int argc, char\* argv[])

{

WSAData wsaData;

WORD DLLVersion = MAKEWORD(2, 1);

system("color 0a");

std::string name = " ";

if (WSAStartup(DLLVersion, &wsaData) != 0)

{

std::cout << "Error" << std::endl;

exit(1);

}

char ip[256];

int port;

std::cout << "///////////" << std::endl;

std::cout << "ENTER IP: ";

std::cin >> ip;

std::cout << "\n///////////" << std::endl;

std::cout << "ENTER PORT: ";

std::cin >> port;

std::cout << "\n///////////" << std::endl;

std::cout << "ENTER NAME: ";

std::cin >> name;

SOCKADDR\_IN addr;

int sizeofaddr = sizeof(addr);

addr.sin\_addr.s\_addr = inet\_addr(ip);

addr.sin\_port = htons(port);

addr.sin\_family = AF\_INET;

Connection = socket(AF\_INET, SOCK\_STREAM, NULL);

if (connect(Connection, (SOCKADDR\*)&addr, sizeof(addr)) != 0)

{

std::cout << "Error: failed connect to server >~<.\n";

return 1;

}

std::cout << "Connected!\n";

CreateThread(NULL, NULL, (LPTHREAD\_START\_ROUTINE)ClientHandler, NULL, NULL, NULL);

std::string msg1;

while (true)

{

std::cin >> msg1;

msg1 = name + ": " + msg1;

int msg\_size = msg1.size();

Packet packettype = Chat;

send(Connection, (char\*)&packettype, sizeof(Packet), NULL);

send(Connection, (char\*)&msg\_size, sizeof(int), NULL);

send(Connection, msg1.c\_str(), msg\_size, NULL);

}

system("pause");

return 0;

}

# Практическое занятие №20. Работа с графами.

**Листинг 72.**

Выполнил: Пихтов Виталий Алексеевич

Построение лабиринта по графам и его вывод на экран. Использовались библиотеки SFML, RANDOM, CTIME, CSTDLIB.

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <random>

#include <SFML/Graphics.hpp>

using namespace std;

const int wallNoEnter = 0, passEnter = 1;

int A, B, countA, countB, idVolnA, idVolnB;

//////////////////////////////

// Функция определяющая тупики

bool deadend(int x, int y, int\*\* mazeArray, int height, int width)

{

int a = 0;

if (x != 1)

{

if (mazeArray[y][x - 2] == passEnter)

a += 1;

}

else a += 1;

if (y != 1)

{

if (mazeArray[y - 2][x] == passEnter)

a += 1;

}

else a += 1;

if (x != width - 2)

{

if (mazeArray[y][x + 2] == passEnter)

a += 1;

}

else a += 1;

if (y != height - 2)

{

if (mazeArray[y + 2][x] == passEnter)

a += 1;

}

else

a += 1;

if (a == 4)

return 1;

else

return 0;

}

/////////////////////////

// Отображение результата

void visualSFML(int\*\* mazeArray, int height, int width)

{

sf::RenderWindow window(sf::VideoMode(640, 480), "Labirint");

sf::Event e;

sf::RectangleShape rectangle;

rectangle.setScale(8, 8);

for (int i = 0; i < A; i += 21)

{

idVolnA++;

}

for (int i = 0; i < B; i += 21)

{

idVolnB++;

}

std::cout << idVolnA << " " << idVolnB << std::endl;

sf::Texture wallTexture;

sf::Sprite wallSprite;

wallTexture.loadFromFile("sprites/wall.png");

wallSprite.setTexture(wallTexture);

wallSprite.setPosition(15, -300);

wallSprite.setScale(0.7, 0.7);

while (window.isOpen())

{

while (window.pollEvent(e))

{

if (e.type == sf::Event::Closed)

window.close();

}

window.clear(sf::Color::Black);

for (int i = 0; i < height; i++)

for (int j = 0; j < width; j++)

{

if (mazeArray[i][j] == wallNoEnter)

wallSprite.setPosition(j \* 8, i \* 8);

if (mazeArray[i][j] == passEnter)

continue;

wallSprite.setPosition(j \* 32, i \* 32);

window.draw(wallSprite);

}

window.display();

}

}

////////////////////////////////

// Алгоритм построения лабиринта

void mazemake(int\*\* mazeArray, int height, int width)

{

int x, y, c, a;

bool b;

for (int i = 0; i < height; i++)

for (int j = 0; j < width; j++)

mazeArray[i][j] = wallNoEnter;

// Координаты и счетчик

x = 3;

y = 3;

a = 0;

while (a < 10000)

{

mazeArray[y][x] = passEnter;

a++;

while (1)

{

c = rand() % 4;

// По две клетки в одном направлении за прыжок

switch (c)

{

case 0: if (y != 1)

if (mazeArray[y - 2][x] == wallNoEnter)

{

mazeArray[y - 1][x] = passEnter;

mazeArray[y - 2][x] = passEnter;

y -= 2;

}

case 1: if (y != height - 2)

if (mazeArray[y + 2][x] == wallNoEnter)

{

mazeArray[y + 1][x] = passEnter;

mazeArray[y + 2][x] = passEnter;

y += 2;

}

case 2: if (x != 1)

if (mazeArray[y][x - 2] == wallNoEnter)

{

mazeArray[y][x - 1] = passEnter;

mazeArray[y][x - 2] = passEnter;

x -= 2;

}

case 3: if (x != width - 2)

if (mazeArray[y][x + 2] == wallNoEnter)

{

mazeArray[y][x + 1] = passEnter;

mazeArray[y][x + 2] = passEnter;

x += 2;

}

}

if (deadend(x, y, mazeArray, height, width))

break;

}

if (deadend(x, y, mazeArray, height, width)) // Вытаскивание из тупика

do

{

x = 2 \* (rand() % ((width - 1) / 2)) + 1;

y = 2 \* (rand() % ((height - 1) / 2)) + 1;

} while (mazeArray[y][x] != passEnter);

}

}

int main()

{

srand((unsigned)time(NULL));

int height = 21, width = 21;

int\*\* mazeArray = new int\* [height];

for (int i = 0; i < height; i++)

mazeArray[i] = new int[width];

mazemake(mazeArray, height, width);

visualSFML(mazeArray, height, width);

for (int i = 0; i < height; i++)

delete[] mazeArray[i];

delete[] mazeArray;

system("pause");

return 0;

}