**Лабораторная работа №15**

**Тема: Выбор и обоснование выбора среды разработки программы. Изучение различных стилей программирования правил формирования листинга программы.**

Задание 1

**Python** стал очень популярным языком программирования. Он используется как для веб-разработки, так и для создания настольных приложений. В Сети можно найти огромное число обучающих сайтов, учебных пособий и руководств по Python, что делает его доступным языком для новичков.

Кроме того, синтаксис Python прост и лёгок для понимания по сравнению с другими языками. Python поддерживает объектно-ориентированное, функциональное и аспектно-ориентированное программирование, а также это динамически типизированный язык с открытым исходным кодом.

### Что можно делать на Python

Некоторые из наиболее популярных применений Python:

* кроссплатформенные shell-скрипты;
* быстрая автоматизация;
* веб-разработка:
* Data Science, Machine Learning.

**HTML** – это язык web-браузеров, с помощью которых созданы сайты. В свою очередь **CSS** заставляет их выглядеть стильно и со вкусом, гораздо лучше, чем такие не очень красивые сайты, действующие с первых дней работы в Интернете.

Преимущества HTML:

1. Сохраняет информацию о выбранных автором шрифтах, кодировках, стилях, цвете текста;
2. Читается на различных компьютерных платформах;
3. Многие программы умеют с ним работать;
4. Открытый стандарт.

Преимущества CSS:

1. **Изменение дизайна веб-страниц становиться проще**;
2. **Поддержка различных устройств;**
3. **Единство дизайна;**
4. **CSS позволяет сохранить время.**

**JavaScript** – это язык интерфейса, используемый для создания и разработки web-сайтов, настольных приложений и игр. JavaScript работает во всех браузерах и может работать с программами, которые не размещены в Интернете. Он также поддерживает как функциональные, так и объектно-ориентированные стили программирования. Компоненты, структуры данных и алгоритмы JavaScript применяются практически к любому другому языку.

Преимущества:

1. **JavaScript-приложение не требует установки на компьютер пользователя**;
2. Скорость работы и производительность
3. **Язык высокого уровня**;
4. **Единство дизайна**;
5. **Прост в изучении**.

**MySQL** – является важной частью web-разработки, что позволяет получать конкретные данные из больших, сложных БД. SQL пользуется большим спросом среди крупных компаний, таких как Microsoft.

Преимущества:

1. Обеспечивает хорошую скорость и производительность;
2. богатый функционал;
3. Легко масштабируется и подходит для больших баз данных;
4. Обеспечивает хорошее управление пользователями и множественный контроль доступа;
5. Легко использовать.

Задание 2

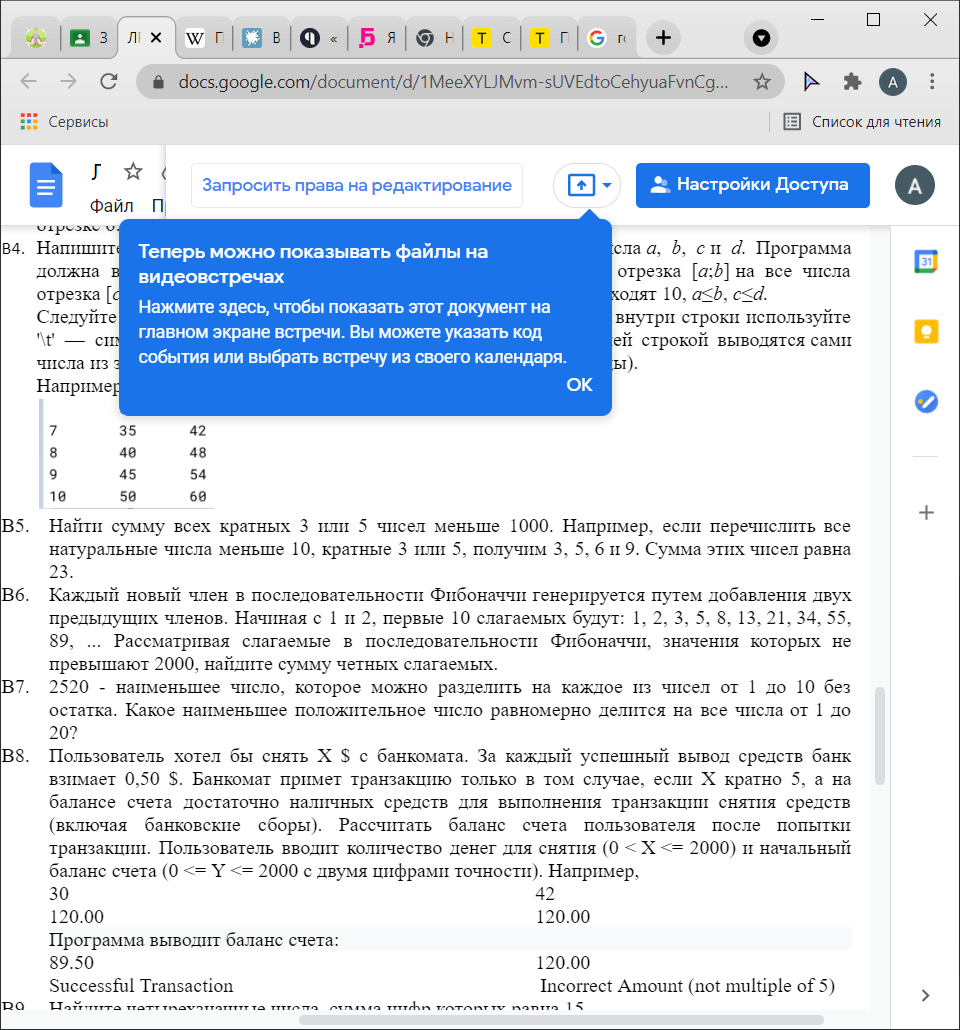
Объектно-ориентированное программирование - Методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования. Сокращается время на разработку, которое может быть отдано другим задачам; компоненты многоразового использования обычно содержат гораздо меньше ошибок, чем вновь разработанные, ведь они уже не раз подвергались проверке. Некоторое снижение производительности функционирования ПО и высокие начальные затраты. Программы, где используются объекты, классы. Обычно такой парадигмой пишут игры

Задание 3

1)Структурное

Разработал: Куклевский Артём Андреевич

По предметной области: Инструментальное ПО.

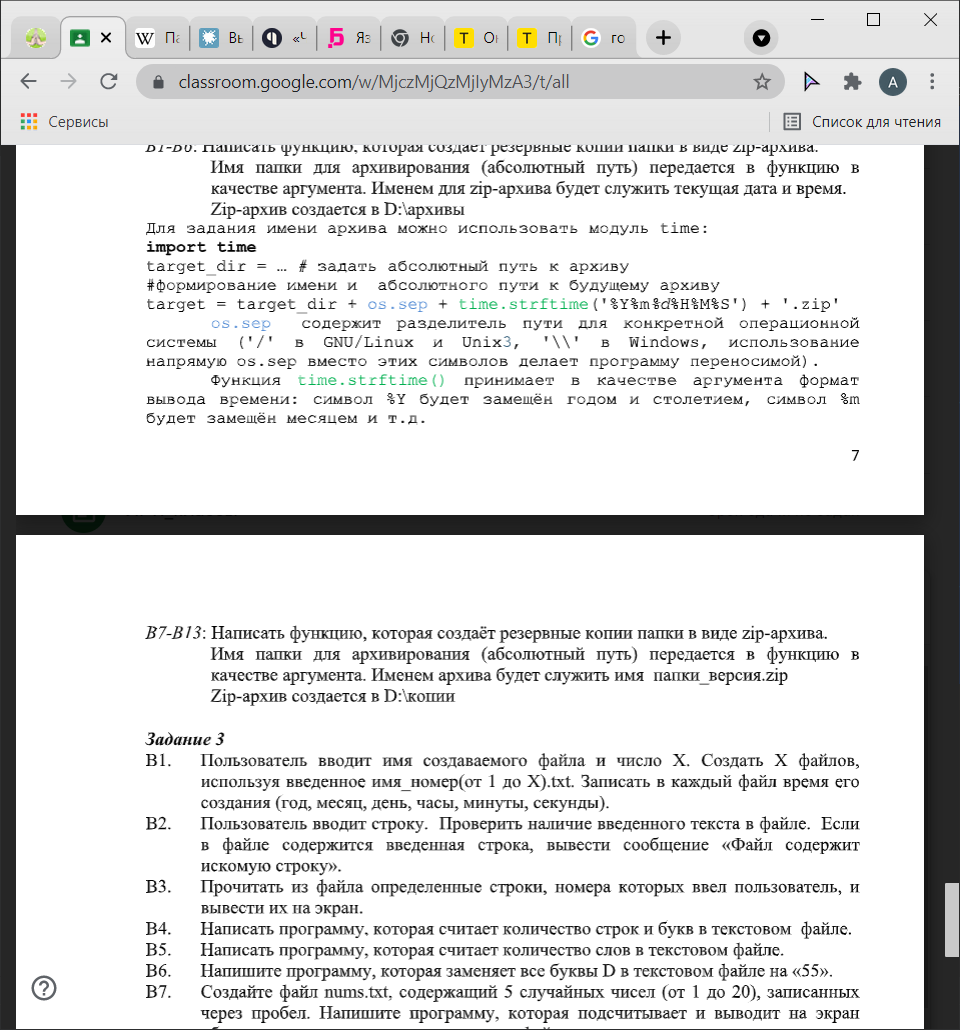


Y = int(input('Введите баланс счёта\n'))  
X = int(input('Введите сколько денег вы хотите снять\n'))  
  
n = int(input('Сколько вы хотите раз снять деньги со счёта?\n'))  
  
if (0 < X <= 2000) and (0 <= Y <= 2000):  
 for i in range(n):  
 if X // 5 == 0:  
 if Y>X:  
 Sum = (Y-X-0.5)  
 Y = Sum  
 print('На вашем счёте осталось денег', Sum,'$ при ',i+1,' снятии')  
 else:  
 print('На вашем счёте не достаточно денег')  
 break  
 else:  
 print('В банкамате только петёрки')  
else:  
 print('Ошибка ввода')

2)Модульное

Разработал: Куклевский Артём Андреевич

По предметной области: Инструментальное ПО.

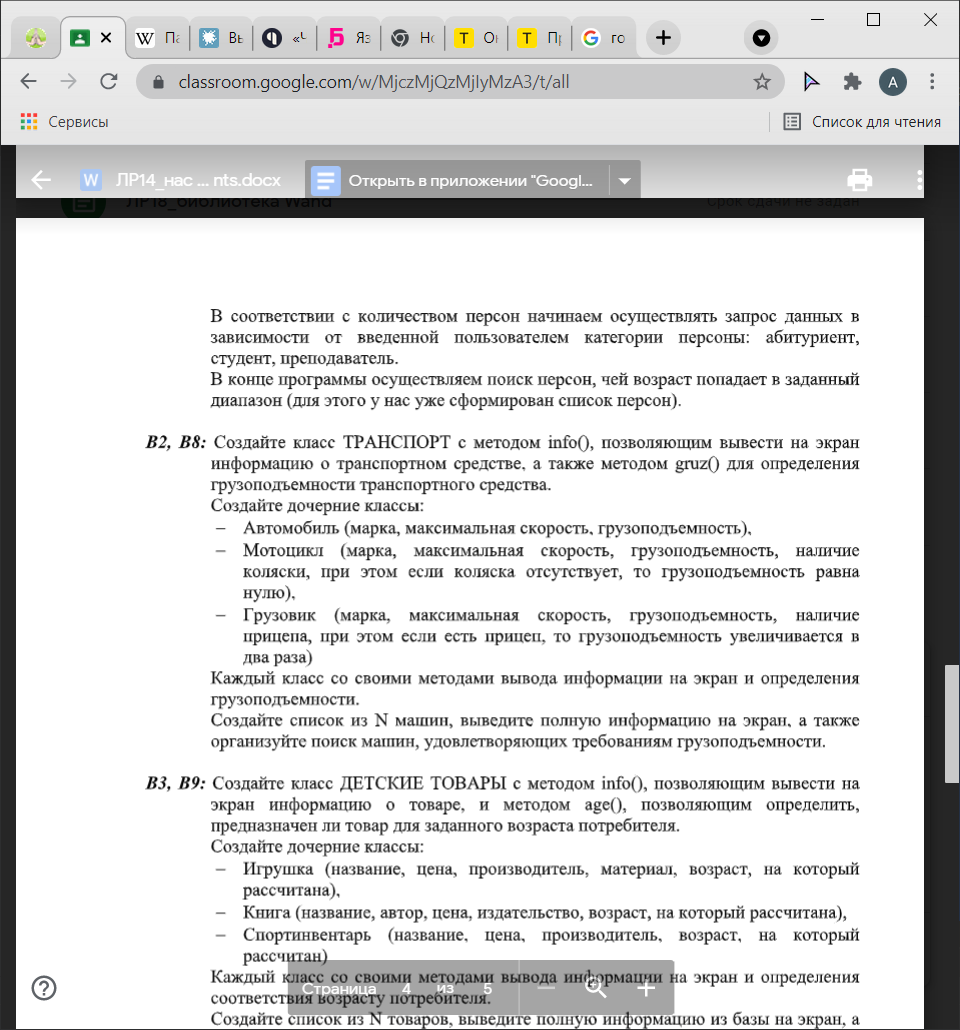


import zipfile, os  
def backupToZip(folder):  
 folder = os.path.abspath(folder)  
 number = 1  
 while True:  
 zipFilename = os.path.basename(folder) + '\_' + str(number) + '.zip'  
 if not os.path.exists(zipFilename):  
 break  
 number = number + 1  
 print('Создаётся файл %s...' % (zipFilename))  
 backupZip = zipfile.ZipFile(zipFilename, 'w')  
 for foldername, subfolders, filenames in os.walk(folder):  
 print('Добавление файлов из папки', foldername)  
 backupZip.write(foldername)  
 for filename in filenames:  
 newBase = os.path.basename(folder) + '\_'  
 if filename.startswith(newBase) and filename.endswith('.zip'):  
 continue  
 backupZip.write(os.path.join(foldername, filename))  
 backupZip.close()  
 print('Готово')  
  
backupToZip('C:\\Users\\Boss\\Desktop\\Python\\Лабы')

3)ООП

Разработал: Куклевский Артём Андреевич

По предметной области: Инструментальное ПО.



class transport:  
 def \_\_init\_\_(self, mark, max\_speed, gruz):  
 self.mark = mark  
 self.max\_speed = max\_speed  
 self.gruz = gruz  
  
 def info(self):  
 print(self.mark, 'Максимальная скорость:', self.max\_speed, 'Грузоподъёмность:', self.gruz, 'кг')  
  
class Aftomabil(transport):  
 def \_\_init\_\_(self, mark, max\_speed, gruz):  
 self.mark = mark  
 self.max\_speed = max\_speed  
 self.gruz = gruz  
  
 def info(self):  
 print(self.mark, 'Максимальная скорость:', self.max\_speed, 'м/с', 'Грузоподъёмность:', self.gruz)  
  
class Motocukle(transport):  
 def \_\_init\_\_(self, mark, max\_speed, gruz, colask):  
 self.mark = mark  
 self.max\_speed = max\_speed  
 self.gruz = gruz  
 self.colask = colask  
  
 def info(self):  
 print(self.mark, 'Максимальная скорость:', self.max\_speed, 'м/с', 'Грузоподъёмность:', self.gruz, 'Коляска:', self.colask)  
  
  
class Gruzovik(transport):  
 def \_\_init\_\_(self, mark, max\_speed, gruz, pricep):  
 self.mark = mark  
 self.max\_speed = max\_speed  
 self.gruz = gruz  
 self.pricep = pricep  
  
 def info(self):  
 print(self.mark, 'Максимальная скорость:', self.max\_speed, 'м/с', 'Грузоподъёмность:', self.gruz, 'Коляска:', self.pricep)  
  
  
garage = []  
quantity = int(input('Сколько транспорта вы хотите ввести: '))  
for i in range(quantity):  
 s1 = input('Введите марку: ')  
 s2 = input('Введите максимальную скорость: ')  
 s3 = int(input('Введите грузоподёмность: '))  
 a = transport(s1, s2, s3)  
 garage.append(a)  
 a.info()  
print()  
quantity = int(input('Сколько автомобилей вы хотите ввести: '))  
for i in range(quantity):  
 s1 = input('Введите марку: ')  
 s2 = input('Введите максимальную скорость: ')  
 s3 = int(input('Введите грузоподёмность: '))  
 b = Aftomabil(s1, s2, s3)  
 garage.append(b)  
 b.info()  
 print()  
quantity = int(input('Сколько мотоциклов вы хотите ввести: '))  
for i in range(quantity):  
 s1 = input('Введите марку: ')  
 s2 = input('Введите максимальную скорость: ')  
 s4 = input('Есть коляска?\n')  
 if s4 == 'нет':  
 s3 = 0  
 else:  
 s3 = int(input('Введите грузоподёмность: '))  
 c = Motocukle(s1, s2, s3, s4)  
 garage.append(c)  
 c.info()  
print()  
quantity = int(input('Сколько грузовиков вы хотите ввести: '))  
for i in range(quantity):  
 s1 = input('Введите марку: ')  
 s2 = input('Введите максимальную скорость: ')  
 s3 = int(input('Введите грузоподёмность: '))  
 s4 = input('Есть прицеп?\n')  
 if s4 == 'нет':  
 pass  
 else:  
 s3 \*= 2  
 v = Gruzovik(s1, s2, s3, s4)  
 garage.append(v)  
 v.info()  
print()  
gruz = int(input('Введите грузоподъёмность\n'))  
for i in range(len(garage)):  
 if garage[i].gruz == gruz:  
 print(garage[i].mark, 'Максимальная скорость:', garage[i].max\_speed, 'Грузоподёмность:', garage[i].gruz)