ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт информатики, математики и робототехники

Кафедра математического и компьютерного моделирования

**Лабораторная работа №1: Работа с форматом INI в Python**

**ОБУЧАЮЩЕГОСЯ**

4 курса группы ПИ-4ИВТ221Б

Санникова Михаила Александровича

|  |  |
| --- | --- |
| Уровень высшего образования:  Направление подготовки (специальность)  Направленность (профиль)  программы  Дата выполнения | высшее образование – бакалавриат  09.03.03 “Прикладная информатика”  Информационные и вычислительные технологии  06.10.2025 |
|  |  |
|  |  |

УФА – 2025

**Постановка задачи:**

Создать файл config.ini

Написать функцию для чтения файла

Добавить в файл config.ini:

• Параметр timeout = 30 в секцию settings

• Новую секцию Paths с параметром home = /usr/local/app

Написать функцию, которая проверяет:

• Существование обязательных параметров (host, user в секции Database)

• Корректность типов данных (host — целое число, debug — булево значение)

**Практическая часть:**

Листинг кода с комментариями

*import* configparser  
*import* os  
*from* functools *import* wraps  
  
  
*def* main():  
 file\_path = 'config.ini'  
  
 create\_default\_config\_file(file\_path)  
 modify\_config\_file(file\_path)  
  
 read\_config\_file(file\_path)  
  
  
*def* create\_default\_config\_file(file\_path):  
 config = configparser.ConfigParser()  
  
 config["database"] = {  
 "host": "localhost",  
 "port": "5432",  
 "user": "admin"  
 }  
 config["settings"] = {  
 "debug": "True",  
 "log\_level": "INFO"  
 }  
  
 *with open*(file\_path, "w", encoding="utf-8") *as* f:  
 config.write(f)  
  
  
*def* modify\_config\_file(file\_path):  
 config = configparser.ConfigParser()  
 config.read(file\_path, encoding="utf-8")  
  
 config["settings"]["timeout"] = "30"  
 config["Paths"] = {  
 "home": "/usr/local/app"  
 }  
  
 *with open*(file\_path, "w", encoding="utf-8") *as* f:  
 config.write(f)  
  
  
*def* check\_file\_exists(func):  
 @wraps(func)  
 *def* wrapper(file\_path):  
 *if not* os.path.exists(file\_path):  
 *print*(f"Не существует файла с заданным путем: {file\_path}")  
 *return*  
 *return* func(file\_path)  
  
 *return* wrapper  
  
  
@check\_file\_exists  
*def* read\_config\_file(file\_path):  
 config = configparser.ConfigParser()  
 config.read(file\_path, encoding="utf-8")  
  
 *if not* is\_data\_valid(config):  
 *print*('Данные некорректны')  
 *return*  
  
 *for* section *in* config.sections():  
 *print*(f"\nСекция [{section}]")  
 *print*("-" \* 30)  
 *for* key, value *in* config.items(section):  
 *print*(f"{key} = {value}")  
 *print*("-" \* 30)  
  
  
*def* is\_data\_valid(config):  
 *return* (  
 config["database"].get("host")  
 *and* config["database"].get("user")  
 *and* config["database"]["port"].isdigit()  
 *and* config["settings"]["debug"].capitalize() *in* ('True', 'False')  
 )  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 main()

Скриншоты выполнения программы

