interface PrizeLogger { // 2 завдання

  (message: string): void;

}

interface Person { // 3 завдання (1)

  name: string;

  email: string;

}

interface Author extends Person { // 3 завдання (2)

  numBooksPublished: number;

}

interface Librarian extends Person { // 3 завдання (3)

  department: string;

  assistCustomer(custName: string): void;

}

interface Employee { // 1 завдання

  id: number;

  name: string;

  surname: string;

  available: boolean;

  salary: number;

  markPrize: PrizeLogger;

}

// Клас UniversityLibrarian, який реалізує інтерфейс Librarian

class UniversityLibrarian implements Librarian { // 4 завдання

  name: string;

  email: string;

  department: string;

  constructor(name: string, email: string, department: string) {

    this.name = name;

    this.email = email;

    this.department = department;

  }

  assistCustomer(custName: string): void {

    console.log(`${this.name} is assisting ${custName}`);

  }

}

const employees: Employee[] = [ // 1 завдання, 2 завдання

  { id: 1, name: 'Ivan', surname: 'Ivanov', available: true, salary: 1000, markPrize: console.log },

  { id: 2, name: 'Petro', surname: 'Petrov', available: true, salary: 1500, markPrize: console.log },

  { id: 3, name: 'Vasyl', surname: 'Vasyliev', available: false, salary: 1600, markPrize: console.log },

  { id: 4, name: 'Evgen', surname: 'Zhukov', available: true, salary: 1300, markPrize: console.log },

];

function getAllEmployees(): Employee[] { // 1 завдання

  return employees;

}

// Функція для отримання робітника за ID

function getEmployeeByID(id: number): Employee | undefined { // 1 завдання

  return employees.find((employee) => employee.id === id);

}

// Функція для виведення інформації про робітника

function printEmployee(employee: Employee) { // 1 завдання

  console.log(`${employee.name} ${employee.surname} got salary ${employee.salary}`);

}

// Виклики функцій до 1 завдання

const allEmployees = getAllEmployees();

console.log('All Workers:', allEmployees);

const employeeById = getEmployeeByID(3);

console.log('Worker by ID:', employeeById);

printEmployee(allEmployees[0]);

// Функція, яка задовольняє інтерфейсу PrizeLogger

const logPrize: PrizeLogger = (message: string) => {

  console.log(`Prize log: ${message}`);

};

// Виклик функції logPrize

logPrize('Congratulations! You won the prize!');

// Оголошення об'єктів

const favoriteAuthor: Author = { // 3 завдання (4)

  name: 'George Orwell',

  email: 'george.orwell@google.com',

  numBooksPublished: 13,

};

// Оголошення та ініціалізація об'єкта favoriteLibrarian

const favoriteLibrarian: Librarian = new UniversityLibrarian('Maria Korneliuk', 'korneliuk.m@google.com', 'Science'); // 3 завдання (5) і 4 завдання (3)

favoriteLibrarian.assistCustomer('Sofia');

// Абстрактний клас ReferenceItem та його нащадок Encyclopedia

abstract class ReferenceItem { // завдання 5 і підпункти в ньому & завдання 7 та підпункти в ньому

  public title: string;

  protected year: number;

  private \_publisher: string = '';

  static department: string = 'Default Department';

  constructor(newTitle: string, newYear: number) {

    console.log('Creating a new ReferenceItem...');

    this.title = newTitle;

    this.year = newYear;

  }

  get publisher(): string {

    return this.\_publisher.toUpperCase();

  }

  set publisher(newPublisher: string) {

    this.\_publisher = newPublisher;

  }

  abstract printCitation(): void;

  printItem(): void {

    console.log(`${this.title} was published in ${this.year}`);

    console.log(`Department: ${ReferenceItem.department}`);

  }

}

class Encyclopedia extends ReferenceItem { // завдання 6 і підпункти в ньому

  public edition: number;

  constructor(title: string, year: number, edition: number) {

    super(title, year);

    this.edition = edition;

  }

  printCitation(): void {

    console.log(`${this.title} - ${this.year}`);

  }

  printItem(): void {

    super.printItem();

    console.log(`Edition: ${this.edition} (${this.year})`);

  }

}

// const ref = new ReferenceItem('Red Data Book of Ukraine', 1980); // завдання 5 (2)

const refBook = new Encyclopedia('Red Data Book of Ukraine', 1980, 1);

refBook.printItem();

refBook.printCitation();

// Виведення значення publisher у верхньому регістрі

refBook.publisher = 'george orwell'; // 5 завдання (4)

console.log(refBook.publisher);