

Додаток А

Міністерство Освіти і науки України
КПІ ім. Ігоря Сікорського
Кафедра ІПІ

ЗВІТ
з виконання лабораторної роботи № 3
з кредитного модуля
“Основи програмування-2. Методологія програмування”

Варіант № 9

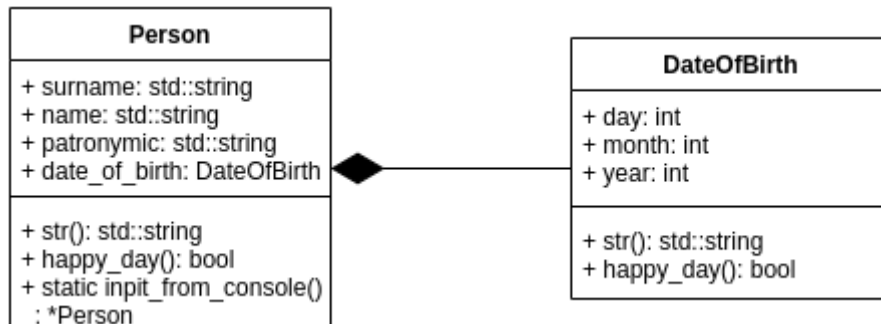
Виконав:
студент 1-го курсу
гр. ІП-25 ФІОТ
Карпов Любомир Васильович

Київ 2023

Постановка задачі

9. Розробити клас "Особа", який характеризується ПІБ особи та її датою народження (у форматі ДД-ММ-РРРР). Створити масив об'єктів даного класу. Визначити людей, які народилися в щасливі дні (ділення суми цифр числа, місяця, року народження на 7 дають три однакові остачі).

Діаграма класів



Текст програми

main.cpp

```
#include <iostream>
#include "Person.h"

int main() {

    int count;
    std::cout << "Enter count of peoples: ";
    std::cin >> count;

    Person *person_array[count];

    for (int i = 0; i < count; ++i) {
        person_array[i] = Person::input_from_console();
    }

    std::cout << "List of peoples:\n";

    for (int i = 0; i < count; ++i) {
        std::cout << person_array[i]->str()
                  << ((person_array[i]->happy_day()) ? " Happy day!" : "")
                  << '\n';
    }
}
```

Person.h

```
#ifndef LAB3_PERSON_H
#define LAB3_PERSON_H

#include <iostream>
#include <utility>

class DateOfBirth {
    int day;
    int month;
    int year;
public:
    DateOfBirth(int day, int month, int year);

    std::string str() const;

    bool happy_day() const;
};

class Person {
    std::string surname;
    std::string name;
    std::string patronymic;
    DateOfBirth date_of_birth;
public:
    Person(
        std::string surname,
        std::string name,
        std::string patronymic,
        DateOfBirth date_of_birth
    ) :
        surname(std::move(surname)),
        name(std::move(name)),
        patronymic(std::move(patronymic)),
        date_of_birth(date_of_birth) {};

    static Person *input_from_console();

    std::string str();

    bool happy_day() { return date_of_birth.happy_day(); };
};

#endif //LAB3_PERSON_H
```

Person.cpp

```
#include "Person.h"

DateOfBirth::DateOfBirth(int day, int month, int year) {
    if (0 < day && day <= 31)
        this->day = day;
    else
        throw "Day must be from 1 to 31";

    if (0 < month && month <= 12)
        this->month = month;
    else
        throw "Month must be from 1 to 12";

    if (1800 <= year && year <= 2023)
        this->year = year;
    else
        throw "Year must be from 1800 to 2023";
}

std::string to_str(int num) {
    std::string s;
    return ((s = std::to_string(num)).size() == 1) ? "0" + s : s;
}

std::string DateOfBirth::str() const {
    return to_str(day)
        + '-' + to_str(month)
        + '-' + to_str(year);
}

int sum_of_digits(int num) {
    int sum = 0;
    while (num > 0) {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}

bool DateOfBirth::happy_day() const {
    int num = sum_of_digits(day) % 7;
    return num == sum_of_digits(month) % 7
        && num == sum_of_digits(year) % 7;
}

std::string input_valid_str(const std::string &input_name) {
    std::string str;

    bool valid = false;
    while (!valid) {
```

```

        std::cout << input_name << ": ";
        getline(std::cin, str);
        valid = true;

        for (char chr: str) {
            if (isdigit(chr))
                valid = false;
        }

        if (!valid)
            std::cout << "Validation error: " << input_name << " is not
valid, try one more time.\n";
    }

    return str;
}

DateOfBirth input_valid_date() {

    while (true) {
        std::cout << "Date of birth (dd-mm-yyyy): ";

        std::string line;
        getline(std::cin, line);
        try {
            int day = std::stoi(line.substr(0, 2));
            int month = std::stoi(line.substr(3, 5));
            int year = std::stoi(line.substr(6, 8));

            try {
                return DateOfBirth(day, month, year);
            }
            catch (const char *exception) {
                std::cout << "Validation error: " << exception << "\n";
            }
        }
        catch (std::invalid_argument) {
            std::cout << "Validation error: date should be in \"dd-mm-yyyy\"
format.\n";
        }
    }
}

Person *Person::input_from_console() {
    std::cin.ignore();

    std::string surname, name, patronymic;
    surname = input_valid_str("Surname");
    name = input_valid_str("Name");
    patronymic = input_valid_str("Patronymic");
}

```

```

    DateOfBirth date_of_birth = input_valid_date();

    return new Person(surname, name, patronymic, date_of_birth);
}

std::string Person::str() {
    return surname + ' ' + name + ' ' + patronymic + ' ' +
date_of_birth.str();
}

```

Результати тестування

```

Enter count of peoples: 3
Surname: Sidorov
Name: Nickolay
Patronymic: Petrovich
Date of birth (dd-mm-yyyy): 10-10-2008
Surname: Holand
Name: Tom
Patronymic: Donalds
Date of birth (dd-mm-yyyy): 28-05-2017
Surname: Beli
Name: Abigale
Patronymic: Rodgers
Date of birth (dd-mm-yyyy): 25-07-2005
List of peoples:
Sidorov Nickolay Petrovich 10-10-2008
Holand Tom Donalds 28-05-2017
Beli Abigale Rodgers 25-07-2005 Happy day!

Process finished with exit code 0

```