

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

Звіт

Про виконання лабораторної роботи № 11

з навчальної дисципліни

“Базові методології та технології програмування”

на тему

“Реалізація програмних засобів оброблення динамічних структур даних
та бінарних файлів”

Виконав

студент академічної

групи Є.А. Красніцька

КБ-22-2

Приймас

викладач кафедри

кібербезпеки та ПЗ

_____ О.Г. Собінов

Лабораторна робота №11

Мета роботи полягає у набутті ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів C++ для керування динамічною пам'яттю та бінарними файловими потоками.

Базові методології та технології програмування 6 Лабораторна робота № 11
<https://github.com/odorenskyi/Burkun-Stanislaw-KI222>
<https://github.com/odorenskyi/Karas-Illia-KI222>
<https://github.com/odorenskyi/Krasnitska-Yelyzaveta-KB222>

ВАРІАНТ 5

— ЗАВДАННЯ НА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ —

Створити базу даних “Відділ кадрів: особова картка працівника, ф. П-2” (дата заповнення, табельний номер, ідентифікаційний номер, стать, вид роботи /основна, за сумісництвом/ ПІБ працівника, дата народження, громадянство, освіта, останнє місце роботи і посада, стаж роботи /днів, місяців, років/, місце проживання, паспортні дані, додаткові відомості, дата і причина звільнення).

За вибором користувача (оператора) програма забезпечує:

- виведення всієї бази на екран або у текстовий файл (на вибір користувача);
- додавання нового запису до бази даних;
- пошук запису за введеним прізвищем працівника;
- вилучення заданого оператором запису з бази даних;
- завершення роботи програми з автоматичним записом БД у файл.

База даних автоматично завантажується з файлу під час запуску програми.

Найменування підприємства (устатков., організації)		Типова форма N П-2		
Код ЄДРПОУ _____		ЗАТВЕРДЖЕНО наказом Держкомстату та Міністерства оборони України від 25 грудня 2009 р. N 495/656		
Дата заповнення	Табельний номер	Індивідуальний ідентифікаційний номер	Стать (чоловіча, жіноча)	Вид роботи (основна, за сумісництвом)
ОСОБОВА КАРТКА ПРАЦІВНИКА				
І ЗАГАЛЬНІ ВІДОМОСТІ				
1. Прізвище _____		Ім'я _____		По батькові _____
2. Дата народження " ____ " ____ 19__ р.		3. Громадянство _____		
4. Освіта (базова загальна середня, повна загальна середня, професійно-технічна, неповна вища, базова вища, повна вища)				
Назва освітнього закладу		Диплом (свідчення), серія, номер		Рік закінчення
Спеціальність (професія) за дипломом (свідченням)		Кваліфікація за дипломом (свідченням)		Форма навчання (денна, вечірня, заочна)

Склад команди IT-проєкта:

1. Карась Ілля
2. Красніцька Єлизавета
3. Буркун Станіслав

План роботи з виконання ІТ-проєкта:

1. Завантажити Git-репозиторій -
<https://github.com/odorenskyi/student-name>;
2. В файл README.md, записати: тему, мету, завдання та варіант.
3. Створити теки Report, prj, Software, TestSuite.
4. В теці \lab11\Report створити Документ Microsoft Word «Звіт до Лабораторної роботи № 11», в ньому написати титульну сторінку, тему, мету, завдання, варіант, склад команди.
5. Розробити детальний аналіз отриманих півзадач та проектування. Для розроблення даних півзадач потрібно використати створений на meeting, заголовковий файл struct_type_project_2.h.
6. Розробити свої дві підзадачі та записати їх в модуль «ModulesПрізвище» підключивши заголовковий файл struct_type_project_2.h, вміст файлу включити до звіту.
7. Завантажити файли на GitHub.
8. Коли всі учасники розроблять свої модулі та проведуть модульні тестування, при успішному тестування, завантажте Git-репозиторії інших учасників.
9. В Code::Blocks IDE створити проєкт консольного додатка prj_2_Прізвище.
10. Відповідно до стандарту ISO/IEC 12207 реалізувати програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проєкта.
11. Провести системне тестування консольного додатка prj_2_Прізвище та свого модуля. При успішному тестування переходи на наступний крок, якщо результат негативний повернутися до додатка, та виправити помилки.
12. Вихідний код (текст) проєкта prj_N_Прізвище включити до звіту та завантажи на Git-репозиторій.

13. Проаналізувати хід виконання лабораторних завдань і самостійно одержані результати, на основі чого сформулювати обґрунтовані висновки з виконаної лабораторної роботи, викласти їх обсягом не менше двох сторінок машинного (комп'ютерного) тексту та включити до звіту.

14. Підготувати й зберегти у \Lab11\Report звіт про виконання лабораторної роботи, оформлений згідно з ДСТУ 3008:2015 “Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання”, зі змістом, визначеним цим порядком виконання лабораторної роботи.

15. Представити до захисту звіт з виконаної лабораторної роботи і проект у Git-репозиторії https://github.com/odorenskyi/student_name.

Аналіз та постановка виконання:

Аналіз виконання Task 11:

Спочатку ми зібралися разом, і почали аналізувати завдання. Потім ми обговорили всі проектні рішення, узгодили архітектуру, загальні алгоритми та інтерфейс модулів. Визначили, що ми будемо працювати з vector, для динамічної структури даних. Потім розділи завдання на підзадачі, щоб кожному учаснику команди потрібно було створити по дві підзадачі, тобто функції.

Прізвище та ім'я	Функції для реалізації
Красніцька Єлизавета	1. Виведення бази даних на екран або у файл. 2. Пошук картки за прізвищем.
Буркун Станіслав	1. Загрузка бази даних. 2. Завершення роботи програми з автоматичним записом бази даних у файл.
Карась Ілля	1. Додавання нової картки в базу даних. 2. Вилучення картки з бази даних.

Постановка виконання Task 11:

Вхідні дані:

1. Функція для виведення бази даних на екран, або у консольне вікно, або у файл.
2. Функція для пошук картки працівника по прізвищу.

Вихідні дані:

1. Виведення бази даних, у консольне вікно або у файл.
2. Виведення картки працівника по введеному прізвищі.

Висновок:

Мета роботи полягає у набутті ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів C++ для керування динамічною пам'яттю та бінарними файловими потоками. Для виконання даної лабораторної роботи №11 потрібно 4 академічні години. Потрібно таке обладнання:

1. Персональний комп'ютер з ОС Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit або Mac OS X;
2. Вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks (www.codeblocks.org) для платформи Windows XP/Vista/7/8.x/10, Linux 32-bit/64-bit або Mac OS X;
3. Текстовий редактор (OpenOffice Writer, Microsoft Word або ін.);
4. Git-репозиторій <https://github.com/odorenskyi/Isachenkov-Eduard-KB-21>;
5. файл-шаблон тестового набору Artifact_TEST_SUITE_lab.doc.

Я дотримувався такого плану роботи:

Спочатку я завантажив свій Git-репозиторій та переглянув нове завдання в папці tasks.

Потім створив Документ Word, та в нього записав мету, номер варіанту, завдання, складу команди та титульну сторінку. Також в текстовий файл README, записав тему, мету, варіант та завдання.

Потім ми зібралися командою(Костянтин, Ян, Едуард) та розпочали мітинг, в ньому ми обговорили такі теми, як:

- Аналіз задачі та вимог до неї.
- Обговорили специфікації ПЗ, концептуальні проектні рішення, сформулювати й узгодити архітектуру програмного засобу, загальні алгоритми функціонування та інтерфейси модулів тощо;
- Визначили вид динамічної структури для реалізації бази даних ПЗ, ми обрали незвичайний тип Vector.
- Створили заголовковий файл `struct_type_project_2.h`, в ньому описали структура бази даних.
- Розподілили підзадачі(функції), кожний учасник команди отримав по дві підзадачі. Я отримав підзадачу, яка виводить всю базу даних в консоль або в файл, також я отримав підзадачу, яка шукає картку працівника по прізвищу та виводить її.
- Склали план роботи з виконання ІТ-проекта, виходячи із розподілених між учасниками команди субзадач та вимог міжнародного стандарту ISO/IEC 12207 щодо виконання процесів реалізації програмних засобів;
- Розроблений план виконання ІТ-проекта затвердити у викладача.

Потім ми почали роботи підзадачі самостійно.

Коли всі вже зробили свої підзадачі, кожен учасник команди завантажив модулі інших учасників команд. Я завантажив їх в папку `prj`.

В Code::Blocks IDE створив проєкт консольного додатка `prj_2_Isachenkov`.

Відповідно до стандарту ISO/IEC 12207 реалізував програмний засіб згідно з завданням, використовуючи функції (модулі), розроблені учасниками команди ІТ-проекта.

Завантажив всі файли на Git-репозиторій.

Отже, закінчивши роботу, я зрозумів, що командна робота – це круто. Ви ділите велику задачу, на маленькі підзадачі, які потім окремо реалізовуєте

та потім в кінці роботи їх з'єднуєте і цей настрій, це відчуття прекрасний. Ви розумієте, що ви зробили пазл, який потім з'єднали в одну красиву картину. Ця робота мені сподобалась.

Додаток №1:

Вихідний код заголовкового файлу struct_type_project_2:

```
#ifndef STRUCT_TYPE_PROJECT_2_H_INCLUDED
#define STRUCT_TYPE_PROJECT_2_H_INCLUDED

using namespace std;

struct date{
    short day = 0;        //1...
    short month = 0;      //1...
    short year = 0;
};

struct entry {
    date creationDate;
    int regNum = -1;
    int id = -1;
    char gender = 'm';
    char typeOfWork = 'f';
    string fullName = "";
    date birthDate;
    string citizenship = "";
    enum {basicSecondary, compSecondary, vocational,
        incompHigher, basicHigher, higher} education[4];
    string placeOfEducation[4] = {"", "", "", ""};
    string lastPlaceOfWork = "";
    string lastJob = "";
    struct {
        short days = 0;
        short months = 0;
        short years = 0;
    } workingExp;
    string placeOfLiving = "";
    string passportInfo = "";
    string additionalInfo = "";
    date dateOfFiring;
    string reasonOfFiring = "";
};

#endif // STRUCT_TYPE_PROJECT_2_H_INCLUDED
```

Вухідний код модуля ModulesKrasnitska:

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <cstring>

#include "struct_type_project_2.h"

using namespace std;

void printPersonalCard(entry personalCard, int i){
    cout << "/" << setw(6) << i+1 << "/" << setw(25) << left << "Дата створення:" <<
    "/" << setw(45)
        << (personalCard.creationDate.day < 10 ? "0" : "") +
    to_string(personalCard.creationDate.day) + "." +
        (personalCard.creationDate.month < 10 ? "0" : "") +
    to_string(personalCard.creationDate.month) + "." +
        to_string(personalCard.creationDate.year) << "/" << endl <<
    resetiosflags(ios::left);
    int edc = 0;
    for(int k = 0; k < 17; k++){
        cout << "/" << setw(7) << "/" << setw(25) << left;
        switch(k) {
            case 0:
                cout << "Реєстраційний номер:" << "/" << setw(45) << personalCard.regNum;
                break;
            case 1:
                cout << "Ідентифікаційний номер:" << "/" << setw(45) << personalCard.id;
                break;
            case 2:
                cout << "Стать:" << "/" << setw(45) << (personalCard.gender == 'm' ?
                "чоловіча" : "жіноча");
                break;
            case 3:
                cout << "Вид роботи:" << "/" << setw(45) << (personalCard.typeOfWork ==
                'f' ? "основна" : "за суміцнством");
                break;
            case 4:
                cout << "ПІБ працівника:" << "/" << setw(45) << personalCard.fullName;
                break;
            case 5:
                cout << "Дата народження:" << "/" << setw(45) <<
                (personalCard.birthDate.day < 10 ? "0" : "") +
                to_string(personalCard.birthDate.day) + "." +
                (personalCard.birthDate.month < 10 ? "0" : "") +
                to_string(personalCard.birthDate.month) + "." +
                to_string(personalCard.birthDate.year);
                break;
            case 6:
                cout << "Громадянство:" << "/" << setw(45) << personalCard.citizenship;
                break;
            case 7:
                cout << "Освіта " + to_string(edc+1) + ":" << "/" << setw(45) <<
                (personalCard.placeOfEducation[edc] == "" ? "-" :
                (personalCard.education[edc] ==
                entry::basicSecondary ? "базова середня" :
                personalCard.education[edc] ==
                entry::compSecondary ? "повна середня" :

```



```

entry::vocational ? "професійно-технічна" :
personalCard.education[edc] ==
entry::incompHigher ? "неповна вища" :
personalCard.education[edc] ==
entry::basicHigher ? "базова вища" : "вища"));
break;
case 8:
cout << " " << "/" << setw(45) << personalCard.placeOfEducation[edc];
for(int j = 0; j < 4; j++){
if (personalCard.placeOfEducation[++edc] != "") {
j -= 2; // repeat writing educations` info
}
}
break;
case 9:
cout << "Останнє місце роботи:" << "/" << setw(45) <<
personalCard.lastPlaceOfWork;
break;
case 10:
cout << resetiosflags(ios::left) << "носада: " << left << "/" << setw(45)
<< personalCard.lastJob;
break;
case 11:
cout << "Стаж роботи:" << "/" << setw(45) << (personalCard.workingExp.days
!= 0 ?
to_string(personalCard.workingExp.days) + " днів " : "") +
(personalCard.workingExp.months != 0 ?
to_string(personalCard.workingExp.months) + " місяців " : "") +
(personalCard.workingExp.years != 0 ?
to_string(personalCard.workingExp.years) + " років " : "");
break;
case 12:
cout << "Місце проживання:" << "/" << setw(45) <<
personalCard.placeOfLiving;
break;
case 13:
cout << "Паспортні дані:" << "/" << setw(45) << personalCard.passportInfo;
break;
case 14:
cout << "Додаткові відомості:" << "/" << setw(45) <<
personalCard.additionalInfo;
break;
case 15:
cout << "Дата звільнення:" << "/" << setw(45) <<
(personalCard.dateOfFiring.day != 0 &&
personalCard.dateOfFiring.month != 0 &&
personalCard.dateOfFiring.year != 0 ?
(personalCard.dateOfFiring.day < 10 ? "0" : "") +
to_string(personalCard.dateOfFiring.day) + "." +
(personalCard.dateOfFiring.month < 10 ? "0" : "") +
to_string(personalCard.dateOfFiring.month) +
"." + to_string(personalCard.dateOfFiring.year) : "-");
break;
case 16:
cout << resetiosflags(ios::left) << "причина: " << left << "/" <<
setw(45)
<< personalCard.reasonOfFiring;
break;
}
cout << "/" << endl << resetiosflags(ios::left);
}

```

```

}

void searchPersonalCard(vector<entry> &database, string surname){
    string LastName;
    vector<int> result;
    cout << "+" << setfill('-') << setw(79) << "+" << endl << setfill(' ')
        << "|" << setw(45) << "КАРТКИ З ПРИЗВИЩЕМ " << setw(33) << left << surname <<
    resetiosflags(ios::left) << "|" << endl
        << "+" << setfill('-') << setw(79) << "+" << endl << setfill(' ');
    for(int i = 0; i < database.size(); i++){
        istringstream(database[i].fullName) >> LastName;
        if(LastName == surname){
            result.push_back(i);
        }
    }
    for(int i = 0; i < result.size(); i++){
        printPersonalCard(database[result[i]], result[i]);
        if(i+1 == result.size())
            cout << "+" << setfill('-') << setw(79) << "+" << endl << setfill(' ');
        else
            cout << "+" << setfill('-') << setw(7) << "+" << setw(26) << "+" << setw(46)
    << "+" << setfill(' ') << endl;
    }
}

void showDatabase(vector<entry> &database){
    ofstream fout;
    #define OUT(option) ((option) == 'c' ? cout : fout)

    char option;
    do {
        cout << "+=====+" <<
endl
        << " Ви хочете вивести базу даних на консоль (c) або у файл (t) - ";
        cin >> option;
        cout << "+=====+" <<
endl;
    }while (option != 'c' && option != 't' && option != 'C' && option != 'T' && option
!= 'e' && option != 'E');
    do {
        string filename = "";
        if (option == 't' || option == 'T' || option == 'e' || option == 'E') {
            cout << " Введіть назву файлу: ";
            cin >> filename;
            cout
    <<
"+=====+" << endl;
            fout.open(filename, ios::out);
        }
        if (option == 'c' || fout.is_open()) {
            OUT(option) << "+" << setfill('-') << setw(79) << "+" << endl << setfill('
')
            << "|" << setw(63) << "ВІДДІЛ КАДРІВ: ОСОБОВА КАРТКА
ПРАЦІВНИКА, Ф. П-2" << setw(16) << "|" << endl
            << "+" << setfill('-') << setw(79) << "+" << endl << setfill('
');
            for (int i = 0; i < database.size(); i++) {
                OUT(option) << "|" << setw(6) << i+1 << "|" << setw(25) << left <<
"Дата створення:" << "|" << setw(45)
                << (database[i].creationDate.day < 10 ? "0" : "")
+ to_string(database[i].creationDate.day) + "." +
                (database[i].creationDate.month < 10 ? "0" : "")
+ to_string(database[i].creationDate.month) + "." +

```

```

to_string(database[i].creationDate.year) << "/"
<< endl << resetiosflags(ios::left);
    int edc = 0;
    for (int j = 0; j < 17; j++) {
        OUT(option) << "/" << setw(7) << "/" << setw(25) << left;
        switch(j) {
            case 0:
                OUT(option) << "Реєстраційний номер:" << "/" << setw(45) <<
database[i].regNum;
                break;
            case 1:
                OUT(option) << "Ідентифікаційний номер:" << "/" << setw(45) <<
database[i].id;
                break;
            case 2:
                OUT(option) << "Стать:" << "/" << setw(45) <<
(database[i].gender == 'm' ? "чоловіча" : "жіноча");
                break;
            case 3:
                OUT(option) << "Вид роботи:" << "/" << setw(45) <<
(database[i].typeOfWork == 'f' ? "основна" : "за суміцництвом");
                break;
            case 4:
                OUT(option) << "ПІБ працівника:" << "/" << setw(45) <<
database[i].fullName;
                break;
            case 5:
                OUT(option) << "Дата народження:" << "/" << setw(45) <<
(database[i].birthDate.day < 10 ? "0" : "") +
to_string(database[i].birthDate.day) + "." +
(database[i].birthDate.month < 10 ? "0" : "") +
to_string(database[i].birthDate.month) + "." +
to_string(database[i].birthDate.year);
                break;
            case 6:
                OUT(option) << "Громадянство:" << "/" << setw(45) <<
database[i].citizenship;
                break;
            case 7:
                OUT(option) << "Освіта " + to_string(edc+1) + ":" << "/" <<
setw(45) << (database[i].placeOfEducation[edc] == "" ? "-" :
(database[i].education[edc] == entry::basicSecondary ? "базова середня" :
database[i].education[edc] == entry::compSecondary ? "повна середня" :
database[i].education[edc] == entry::vocational ? "професійно-технічна" :
database[i].education[edc] == entry::incompHigher ? "неповна вища" :
database[i].education[edc] == entry::basicHigher ? "базова вища" : "вища"));
                break;
            case 8:
                OUT(option) << " " << "/" << setw(45) <<
database[i].placeOfEducation[edc];
                if (database[i].placeOfEducation[++edc] != "") {
                    j -= 2; // repeat writing educations` info
                }
                break;
            case 9:
                OUT(option) << "Останнє місце роботи:" << "/" << setw(45) <<
(database[i].LastPlaceOfWork == ""

```

```

?
    "-" : database[i].lastPlaceOfWork);
        break;
    case 10:
        OUT(option) << resetiosflags(ios::left) << "носада:      " <<
left << "/" << setw(45) << (database[i].lastJob == "")
? "-" : database[i].lastJob);
        break;
    case 11:
        OUT(option) << "Стаж роботи:" << "/" << setw(45) <<
(database[i].workingExp.days != 0 &&
database[i].workingExp.months != 0 &&
database[i].workingExp.years != 0 ?
to_string(database[i].workingExp.days) + " днів " : "") +
(database[i].workingExp.months != 0 ?
to_string(database[i].workingExp.months) + " місяців " : "") +
(database[i].workingExp.years != 0 ?
to_string(database[i].workingExp.years) + " років " : "") : "-");
        break;
    case 12:
        OUT(option) << "Місце проживання:" << "/" << setw(45) <<
database[i].placeOfLiving;
        break;
    case 13:
        OUT(option) << "Паспортні дані:" << "/" << setw(45) <<
database[i].passportInfo;
        break;
    case 14:
        OUT(option) << "Додаткові відомості:" << "/" << setw(45) <<
(database[i].additionalInfo == "")
?
    "-" : database[i].additionalInfo);
        break;
    case 15:
        OUT(option) << "Дата звільнення:" << "/" << setw(45) <<
(database[i].dateOfFiring.day != 0 &&
database[i].dateOfFiring.month != 0 &&
database[i].dateOfFiring.year != 0 ?
to_string(database[i].dateOfFiring.day) + "." +
to_string(database[i].dateOfFiring.month) +
to_string(database[i].dateOfFiring.year) : "-");
        break;
    case 16:
        OUT(option) << resetiosflags(ios::left) << "причина:      "
<< left << "/" << setw(45)
<< (database[i].reasonOfFiring == "" ? "-" :
database[i].reasonOfFiring);
        break;
    }
    OUT(option) << "/" << endl << resetiosflags(ios::left);
}
if (i+1 != database.size())
    OUT(option) << "+" << setfill('-') << setw(7) << "+" << setw(26)
<< "+" << setw(46) << "+" << setfill(' ') << endl;
else
    OUT(option) << "+" << setfill('-') << setw(79) << "+" << endl <<
setfill(' ');

```

```

    }
    if (fout.is_open()) fout.close();
    break;
}
else {
    cerr << "Сталася помилка під час написання " << filename << ". Спробуйте
ще раз." << endl;
}
}while (true);

#undef OUT(option)
}

```

Вихідний код заголовкового файлу Modules Krasnitska:

```

#ifndef DATABASESTRUCT_H_INCLUDED
#define DATABASESTRUCT_H_INCLUDED

#include "struct_type_project_2.h"

void showDatabase(vector<entry> &database);
void searchPersonalCard(vector<entry> &database, string surname);

#endif // DATABASESTRUCT_H_INCLUDED

```

Додаток №2:

Системне тестування prj_2_Krasnitska файл TestSuite №11:

Назва тестового набору Test Suite Description	TestSuite №11
Назва проекту / ПЗ Name of Project / Software	prj_2_Krasnitska.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Красніцька Єлизавета
Виконавець Implementer	Красніцька Єлизавета

Ід-р тес т- кей са / Test Cas e ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результ ат тестува ння (пройшов/не вд алося/ заблокований) / Test Result (passed/fa iled/ blocked)
ТС _1	1. Відкри ти prj_2_Kr asnitska. exe 2. Ввести “ 25, 45, 56” 3. Ввести “ 12.03.2 022”	Введіть номер операції, яку ви бажаєте виконати: >>> 2 Дата заповнення: {день.місяць.рік} >>> 25, 45, 56 Дата заповнена неправильно Дата заповнення: {день.місяць.рік} >>> 12.03.2022 Табельний номер:	Passed

4. Ввести “ 45”	>>> 45	
5. Ввести “ 38”	Ідифікаційний номер: >>> 38	
6. Ввести “ m”	Стать(m - чоловік, w - жінка):	
7. Ввести “ f”	>>> m	
8. Ввести “ Задоро жний Костян тин Олекса ндрови ч”	Вид роботи(f - основна, p - за сумісництвом): >>> f ПІБ працівника: >>> Карась Ілля Миколайович День народження: {день.місяць.рік} >>> 05.01.2005	
9. Ввести “ 22.10.2 004”	Громадянство: >>> Українець	
10.Ввести “ Україн ець”	Введіть кількість освіт(максимальна кількість освіт для введення 4) >>> 2	
11.Ввести “2”	Освіта:	
12.Ввести “ 2 ”	1 - базова середня	
13.Ввести “ГНТН ”	2 - повна середня, 3 - професійна,	
14.Ввести “ 4”	4 - неповна вища, 5 - базова вища,	
15.Ввести “ЦНТ у”	6 - вища): >>> 2	
16.Ввести “ n ”	В якому закладі здобували освіту? >>> ГНТН	
17.Ввести “ Кропи вницьк ий,	Освіта: 1 - базова середня 2 - повна середня, 3 - професійна,	

	<p>Україна”</p> <p>18.Ввести “</p> <p>Паспортні дані”</p> <p>19.Ввести “</p> <p>Додаткова інформація”</p>	<p>4 - неповна вища,</p> <p>5 - базова вища,</p> <p>6 - вища):</p> <p>>>> 4</p> <p>В якому закладі здобували освіту?</p> <p>>>> ЦНТУ</p> <p>Ви працювали?(Y/y - так N/n - ні)</p> <p>>>> n</p> <p>Місце проживання:</p> <p>>>> Кропивницький, Україна</p> <p>Паспортні дані:</p> <p>>>> Паспортні дані</p> <p>Додаткова інформація:</p> <p>>>> Додаткова інформація</p>	
ТС _2	<p>1. Відкрити prj_2_ Krasnits ka.exe</p> <p>2. Ввести “ 1”</p> <p>3. Ввести “ c ”</p>	<p>Введіть номер операції, яку ви бажаєте виконати:</p> <p>>>> 1</p> <p>+=====</p> <p>=====+</p> <p>Ви хочете вивести базу даних на консоль (c) або у файл (t) - c</p> <p>+=====</p> <p>=====+</p> <p>+=====</p> <p>=====+</p> <p>Ви хочете вивести базу даних на консоль (c) або у файл (t) - c</p> <p>+=====</p> <p>=====+</p> <p>+_____</p> <p>_____+</p>	Passed

ВІДДІЛ КАДРІВ: ОСОБОВА КАРТКА ПРАЦІВНИКА,
Ф. П-2

+

+

1|Дата створення: |11.05.2023

|Реєстраційний номер: |45

|Ідентифікаційний номер: |38

|Стать: |чоловіча

|Вид роботи: |основна

|ПІБ працівника: |Карась Ілля Миколайович |

|Дата народження: |05.01.2005|

|Громадянство: |Українець

|Освіта 1: |базова вища

| |ГНТН

|Освіта 2: |неповна вища

| |ЦНТУ

|Останнє місце роботи: |-

| посада: |-

|Стаж роботи: |-

|Місце проживання: |Кропивницький, Україна

		<div> <div>Паспортні дані:</div> <div>Паспортні дані</div> </div> <div> <div>Додаткові відомості:</div> <div>Додаткова інформація</div> </div> <div> <div>Дата звільнення:</div> <div>-</div> </div> <div> <div>причина:</div> <div>-</div> </div> <div>_____</div> <div>_____+</div>	
ТС _3	<div>1. Відкрити prj_2_Krasnitska.exe</div> <div>2. Ввести "1"</div> <div>3. Ввести "t"</div> <div>4. Ввести «PersonalCard.txt»</div> <div>5. Ввести "5"</div> <div>6. Впапці з програмою відкрити файл «PersonalCard.txt».</div>	<div>Введіть номер операції, яку ви бажаєте виконати:</div> <div>>>> 1</div> <div>+=====</div> <div>=====+</div> <div>Ви хочете вивести базу даних на консоль (c) або у файл (t) - t</div> <div>+=====</div> <div>=====+</div> <div>Введіть назву файлу: PersonalCard.txt</div> <div>+=====</div> <div>=====+</div>	Passed
ТС _4	<div>1. Відкрити prj_2_Krasnitska.exe</div>	<div>Введіть номер операції, яку ви бажаєте виконати:</div> <div>>>> 4</div> <div>Введіть ідентифікаційний номер картки, яку ви хочете вилучити:</div>	Passed

	<div>2. Ввести “4”</div> <div>3. Ввести “445”</div> <div>4. Ввести “4”</div> <div>5. Ввести “38”</div>	<div>>>> 445</div> <div>+-----+</div> <div> Запис не знайдено </div> <div>+-----+</div> <div>Введіть номер операції, яку ви бажаєте виконати:</div> <div>>>> 4</div> <div>Введіть ідентифікаційний номер картки, яку ви хочете вилучити:</div> <div>>>> 38</div> <div>+-----+</div> <div> Запис видалено </div> <div>+-----+</div>	
ТС_5	<div>1. Відкрити prj_2_Krasnitska.exe</div> <div>2. Ввести “ 5 ”</div>		Passed