Міністерство освіти і науки, молоді та спорту України

Національний університет «Львівська політехніка»

Кафедра АСУ



Лабораторна робота №1

з дисципліни: «Інструментальні засоби проектування та розробки автоматизованих систем»

на тему: «**Структура класу та механізми доступу до його компонентів**»

Виконав:

ст. гр. ІУСм-11

Скиба Ростислав

Прийняв:

Ковальчук А.М.

Львів 2016

***Мета роботи:*** Вивчення базових елемент·в ООП (поняття класу, протоколу, конструктора і т.д.) та їх застосування для розв"язку прикладних задач.

1. **ТЕОРЕТИЧНІ ВІДОМОСТІ**

**1.1. *Типи даних і класи.***

Якщо в програмі на мові С++ написано "int nCount", то зро­зуміло, що тип int визначає можину можливих значень змінної nCo­unt. Але поняття типу не зводиться тільки до такого розуміння.

Тип Т варто розуміти як пару Т = <At,Op>, де At визначає множину можливих значень типу, a Op задаєµ множину операцій над значеннями даного типу, в яких можуть також бути і значення інших типів,· які вже µ задані.

Як тільки визначений клас (тип), наприклад з ·менем А, можна говорити про об’єкти цього класу (змінні типу А), які в ОПП називаються ще екземплярами класу.

Розділ атрибутів призначений для оголошення змінних класу. Вони · задають атрибути об’єктів класу. Їх типи в сукупност· задають множину можливих значень, або, станів об’єктів класу.

Розділ операцій призначений для оголошення і визначення операцій, які можна виконувати над об’єктами класу або, як прийнято казати в ООП, виконуваних самими об’єктами. В ООП операції називаються методами класу. Методи описують поводження об’єктів класу. Синтаксично оголошення і визначення операцій (методів) розділено.

**1.2. *Класи та інкапсуляція.***

Отже, іншими словами, клас об’єднує “під одним дахом" дані і операції. Вже таке розміщення даних і операцій в одній "капсулі" можна розглядати як свого роду· інкапсуляцію. Однак під цим терміном розуміють приховані реалізації операцій · даних класу від зовнішних у відношенні до нього програм, які використовують клас.

Щоб реалізувати інкапсуляцію операцій, визначення класу розбивають на дві частини: інтерфейс класу і реалізацію класу.

Інтерфейс класу(або протокол класу містить тільки оголошення його елементів: змінних і методів. Визначення методів у вигляді їх реалізації як функцій мови С відокремлені від інтерфейсу і розміщуються в реалізації класу. Такий підхід дозволяє використовувати клас, знаючи тільки його інтерфейс, не вдаючись в деталі реалізації. З інтерфейсу можна зрозуміти, які в класі операції і як їх викликати.

Реалізація операцій може бути довільною, аби лише вона відповідала семантиці. Зміна реалізації при збереженні інтерфейсу і семантики не впливає на програми, які використовують даний клас.

**1.3. *Конструктори і деструктори.***

Конструктори і деструктори **-** це спеціальні методи класу, які явно або неявно викликаються при створенні або знищенні об’єктів. Кожний клас повинен мати хоч би один конструктор, але, як правило, задаються декілька. Конструктор відіграє дві ролі. Головна **-** створення "пустого" об’єкта, тобто розподіл пам’яті для змінних класу цього об’єкта. Друга **-** ініціалізація стану об’єкта, тобто заповненя "пустого" об’єкта. Саме в конструкторі визначається, якими значеннями потрібно проініціалізувати змінні об’єкта.

Іменем конструктора є ім’я класу. **Конструктор по замовчуванню** не має параметрів, або всі його параметри отримують значення по замовчуванню. Він викликається завжди без параметрів. Якщо у визначенні класу конструктор не оголошений, то автоматично створюється конструктор по замовчуванню з пустим тілом і який виконує головну роль без ініціалізації. Наприклад, в результаті обробки компілятором оголошення об’єкта.

Patient Petrov;

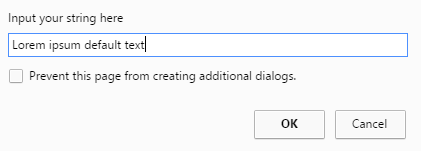
викликається конструктор по замовчуванню **-** або визначений в класі Patient, або створений компілятором.

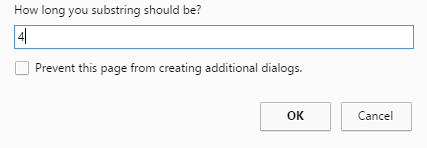
Існують ще два спеціальних типи: **конструктор копії** і **конструктор перетворення типу**. Перший має один параметр **-** константу, або звичайну зсилку на об’єкт даного класу, і дозволяє при ствоворенні об’єкта ініціювати його занченням іншого об’єкта.

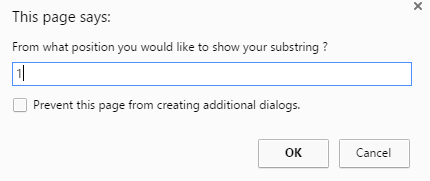
1. **ХІД РОБОТИ**
   1. **Завдання**

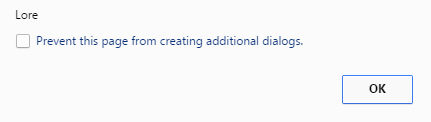
Не використовуючи стандартних процедур та функцiй, написати програму для видiлення пiдрядка заданої довжини iз рядка, починаючи iз за даної позицiї;

* 1. **Результат роботипрограми**









* 1. **Код програми**

**main.js**

**(function () {**

**"use strict"**

**var str,**

**str2 = [],**

**j = 0;**

**str = prompt("Input your string here");**

**var long,**

**i = 0;**

**long = prompt("How long you substring should be?");**

**var index = prompt("From what position you would like to show your s ubstring?");**

**for (index; i <long; index++, i++)**

**{**

**str2[j] = str[index-1];**

**j++;**

**}**

**var str2 = str2.join("");**

**alert (str2);**

**}) ();**

**Висновки:** В даній лабораторній роботі я повторив базові елементи·в ООП (поняття класу, протоколу, конструктора і т.д.) та застосував їх для розв"язку прикладної задач.