**Практична робота № 10 Розроблення плану тестування програм та тестів**. **Підсумкова контрольна робота**

**Мета:** навчитися складати план тестування програм та тестів розробленої програми**.**

**Строк виконання 10.05.2021**

**Завдання.**

1. Повторити лекцію 11 та ознайомитися з теоретичною частиною.
2. Для лабораторної роботи №13 розробіть план тестування, визначивши предмет перевірки та очікуваний результат, заповнивши таблицю:

План тестування

|  |  |  |  |
| --- | --- | --- | --- |
| №№ | Предмет перевірки  (що перевіряємо) | Дії, що виконуються | Очікуваний результат  (результат правильної роботи) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Виконати підсумкову контрольну роботу у вигляді тестування. **Робота надається після теоретичної частини.**
2. Підготуватися до заліку. Запитання до заліку надані за посиланням <https://github.com/omc-college/PTBD21-algorithmization-programming>, а також надані після тестів підсумкової контрольної роботи.

**План тестування та відповіді до контрольної роботи надаються** на електронну адресу викладача [t.i.lumpova@gmail.com](mailto:t.i.lumpova@gmail.com)текстовим файлом з іменем у форматі

**<Назва групи англійською>P10<Прізвище англійською>**.

Наприклад, PTBD21**P10**Ivanov.doc.

Ви можете використати цей файл, перейменувавши його відповідним чином.

**ТЕОРЕТИЧНА ЧАСТИНА**

Програм без помилок не існує. Практика доводить, що винуватцями помилок у програмах найчастіше бувають самі програмісти. Один із загальних законів практичного програмування полягає в тому, що жодна програма не дає бажаних результатів при першій спробі трансляції та виконання. Певне уявлення про справжні причини появи помилок у роботі програми дає таке процентне співвідношення джерел збоїв:

|  |  |
| --- | --- |
| Вхідні дані | 1% |
| Помилки користувача | 5% |
| Апаратура | 1% |
| Системне програмне забезпечення | 3% |
| Розробка системи | 15% |
| Програмування | 75% |

     Програміст повинен не тільки писати ефективні програми, але і знаходити в них усілякі помилки. Сучасна практика навчання програмуванню орієнтована, в основному, тільки на виконання програмістом першої половини своєї роботи. Це все одно, як навчати льотчика тільки зльоту, припускаючи, що з посадкою машини він якось розбереться сам, якщо буде виконувати всі операції зльоту в зворотному порядку.

     Існують два типи програмних помилок:

*синтаксичні помилки* - виникають через порушення правил мови програмування. Такі помилки зазвичай виявляються під час компіляції. Можуть бути виключені порівняно легко. Навіть якщо не переглядати текст програми можна бути впевненим, що компілятор на стадії трансляції знайде помилки і видасть відповідні попередження. Фактично пошук помилок здійснює компілятор, а їхнє виправлення - програміст;

*семантичні (логічні) помилки* - ті, що призводять до некоректних обчислень або помилок під час виконання (run-time error). Семантичні помилки усувають зазвичай за допомогою виконання програми з ретельно підібраними перевірочними даними, для яких відома правильна відповідь.

 Тестування - процес також ітераційний. Після виявлення та виправлення кожної помилки обов'язково слід повторити тести, щоб переконатися у працездатності програми. Більше того, для ідентифікації причини виявленої проблеми може знадобитися проведення спеціального додаткового тестування. При цьому потрібно завжди пам'ятати про фундаментальний висновок, зроблений професором Едсжером Дейкстрою у 1972 році: "Тестування програм може служити доказом наявності помилок, але ніколи не доведе їхню відсутність!".

Ми коротко розглянемо принципи проведення ручного тестування, яке проводить програміст, на відміну від використання автоматизованих засобів тестування. **Суть ручного тестування** — полягає в тому, що програміст або тест-інженер приміряє на себе, неначе костюмчик, роль кінцевого користувача і, відповідно до тестового сценарію, перевіряє програмне забезпечення, проте не без участі логіки застосовує різноманітні техніки тест дизайну, планує, складає тест сценарії. **Головні мотиви такого тестування** — виявити поведінку програми, яка відрізняється від очікуваної, знизити кількість дефектів, забезпечити працездатність програми, оцінити зручність користування, і кінець-кінцем отримати **якісний** продукт, який буде задовольняти потребам користувачів. В програмі не повинно бути дій з невизначеним результатом.

**Підготовчий етап тестування**

На цьому етапі потрібно:

1. Здійснити аналіз вхідних даних і виявити значення, які можуть привести до неочікуваного або аварійного завершення. Наприклад, перевірка знаменника на нульове значення.
2. Визначити перелік функцій, які виконує програма, на зробити план перевірки кожної функцій, зокрема, що в результаті її виконання видається правильний результат. Так, для обчислювальних операцій потрібно провести "ручний" розрахунок даних, з результатами якого потрібно порівнювати отримані за виконання програми. Потрібно зауважити, що при відхиленні отриманого результату від очікуваного, повинно бути пояснення.
3. Перевірка функцій повинна здійснюватися за заздалегідь розробленим планом. Наприклад, програма виконує такий перелік функцій, який надається через меню:
4. Введення даних.
5. Виведення даних на консоль.
6. Виведення введених даних до файлу.
7. Читання файлу.
8. Модифікація даних.
9. Перезапис даних у файлі.
10. Дозапис даних до файлу.
11. Вихід з програми.

Якщо надається можливість обрати режим роботи програми введенням числа, яке відповідає потрібним діям (як у ваших останніх лабораторних роботах), то введення числа, яке не відповідає жодному з очікуваних режимів повинно викликати відповідні дії, наприклад, виведення відповідного повідомлення та повторний запит режиму (уникнення невизначеності).

Іншим прикладом тестової перевірки є визначення дій, які відбуваються, якщо запитано виведення даних на консоль, а дані не були ні введені в інтерактивному режимі, ні прочитані з файлу. Відповідно аналізується результат і, в разі потреби коригується програма.

Ще один приклад, перевірка дій програми в разі спроби прочитати файл що відсутній.

Всі ситуації, які можуть викликати неправильний або небажаний результат повинні бути визначені та оброблені програмістом. Останнє означає, що повинно видаватися відповідне повідомлення щодо виявленої нештатної ситуації (наприклад, введення даних не того типу або не з тими значеннями, що очікується, або відсутність файлу для оброблення).

1. Перевірка допустимості даних, наприклад, при введенні числових даних перевіряти, що дані дійсно числа, а якщо для значення є обмеження, то числа відповідають цим обмеженням; або, якщо очікується відповідь "так" чи "ні", введена інша відповідь.
2. Перевірка на можливість появи нескінченного циклу внаслідок некоректних даних.

Це тільки мінімальний перелік перевірок. Чим складніша програма, тим більше тестових випадків потрібно розглядати.

**ПІДСУМКОВА КОНТРОЛЬНА РОБОТА У ВИГЛЯДІ ТЕСТУ**

Кожна відповідь оцінюється в 0,25 балів. Відповідь вважається вірною, коли відмічені всі правильні варіанти відповіді. Надати відповіді відмітивши заливкою блакитного кольору.

Відмітьте заливкою блакитного кольору правильну відповідь

Якщо у вас нема технічної можливості прислати doc-файл, відповіді на тест можна прислати у вигляді:

**<номер запитання> - <відповідь>; <номер запитання> - <відповідь>; …<номер запитання> - <відповідь>;**

1. **Коментар – це:**

**Оберіть правильне твердження**

1. фрагмент тексту програми, який слугує для пояснення призначення програми чи окремих команд і не впливає на виконання команд.
2. програма, яка приєднує програмний код ;
3. частина програми, де змінна придатна для повноцінного опрацювання;

**2. Вкажіть правильне звернення до змінних структури:**

**struct coord{int x; int z;};**

**coord data;**

**Оберіть відповідь**

A. data.x; data.z;  
B. coord.x; coord.z;  
C. coord.x; data.z;  
D. data.y.

**3. Компілятор – це:**

**Оберіть правильне твердження**

1. програма, яка перекладає вихідний код мови на мову зрозумілу комп’ютеру.
2. програма з символьним типом;
3. пойменована ділянка оперативної пам’яті комп’ютера, де зберігається значення деякої величини;

**4. Дано: struct coord{int x; int z[];};**

**Що означає такий запис struct coord data[100];:**

**Оберіть правильне твердження**

A. оголошується масив, кожен елемент якого – структура типу coord;

B. оголошується структура data, кожен елемент якої – масив.

**5. Що означає директива** **main** ?

**Оберіть правильне твердження**

1. кінець рядка;
2. початок рядка;
3. заголовок функції.

**6. Потрібно визначити масив дійсних чисел, в якому 20 елементів та ім’я massiv. Оберіть відповідь, яка опису цей масив..**

1. int massiv (20);
2. int massiv [20];
3. int massiv {20};
4. float massiv (20);
5. float massiv [20];
6. float massiv {20};
7. char massiv {20};
8. char massiv [20];
9. char massiv (20);

**7. З поданих нижче варіантів не є дійсним типом:**

**Оберіть відповідь**

1. **float;**
2. **double;**
3. **long double;**
4. **long int.**

**8. Який рядок правильно описує змінну txt1, в якій знаходиться символ W?**

**Оберіть відповідь**

* 1. **char txt1='W';**
  2. **char txt1="W";**
  3. **string txt1="W";**
  4. **string txt1='W';**
  5. **char txt1=[W];**
  6. **string txt1=[W];**

**9. Оберіть вірний результат виконання операції: 2\* -5 + 4.**

1. -6.
2. -5;
3. 12;

**10. Який рядок правильно описує змінну txt2, в яку можна записати текст довжиною 25 символів?**

**Оберіть відповідь**

* 1. char txt2[25];
  2. char txt2(25);
  3. text txt2[25];
  4. text txt2(25);
  5. string txt2[25];
  6. string txt2(25);

**11.** **Ознака ios::in**

**Оберіть правильне твердження**

1. якщо файл, який відкривається для записування вже існує, то цього вміст буде вилучено;
2. відкриває файл для читання з нього інформації:
3. забороняє перезаписувати існуючий файл;

**12. Який результат виконання програми?**

**char l1;**

**char tx[30]="qwertyu";**

**for (int i=strlen(tx)-1;i=0;i--)**

**{**

**l1=tx[i];**

**cout << l1;**

**}**

**cout << endl;**

**Оберіть відповідь**

1. серед відповідей нема вірної
2. виводить рядок **qwertyu** на екран
3. виводить рядок **qwertyu** на екран в зворотному порядку
4. виводить рядок **qwertyu** на екран (в кожному рядку екрану 1 символ з **qwertyu** )
5. виводить рядок **qwertyu** на екран в зворотному порядку (в кожному рядку екрану 1 символ з **qwertyu** )

**13. В якому з рядків виконується звернення до сьомого елементу масиву, розмір масиву дорівнює 10?**

**Оберіть відповідь**

A. mas (7);

B. mas [6];

C. mas 8;

D. mas [7];

**14.** **Оберіть відповідь, що буде надруковано в результаті виконання наступного коду:**

1. **int main()**
2. **{**
3. **for (int i = 0; i < 4; ++i)**
4. **{**
5. **switch (i)**
6. **{**
7. **case 0  : std::cout << "0";**
8. **case 1  : std::cout << "1"; continue;**
9. **case 2  : std::cout << "2"; break;**
10. **default : std::cout << "D"; break;**
11. **}**
12. **std::cout << ".";**
13. **}**
14. **return 0;**
15. **}**

**Можливі відповіді:**

1. 0112.D.
2. 0.1.2.
3. 011.2.D
4. Помилка компіляції в рядку 10
5. 01.2.D.
   1. **Оберіть результат роботи наступного фрагменту коду:**

**int x = 0;**

**switch(x)**

**{**

**case 1: cout << "One";**

**case 0: cout << "Zero";**

**case 2: cout << "Hello, World";**

**}**

**Можливі відповіді:**

1. **One**
2. **Zero**
3. **Hello, World**
4. **ZeroHello, World**
5. **Який фрагмент виведе "Hello,my world!" після виконання коду**

Оберіть правильні відповіді

А

string x("Hello,");

string y(" world!");

x.insert(6,"my");

cout << x+y<< endl;

B

string x("Hello,");

string y("my world!");

y.insert(6,x);

cout << х<< endl;

C

string x("Hello,");

string y("my world!");

x.insert(5,y);

cout << x<< endl;

D

string x("Hello,");

string y(",my world!");

x.replace(6,1,y) ;

cout << y<< endl;

17. **Дан масив** **int L[3][3] = {{2,3,4}, {3,4,8}, {1,0,9}};.** **Чому буде дорівнювати значення елементу цього масиву** **L[1][2]**

***Оберіть потрібну відповідь***

1. 2
2. 3
3. 4
4. 8

**18. Вкажіть директиву препроцесора, яку необхідно підключити для організації форматованого введення-виведення даних:**

***Оберіть потрібну відповідь***

1. #include <iostream.h>
2. #include <stdio.h>
3. #include <math.h>
4. #include <conio.h>

**19. Ідентифікатор – це:**

***Оберіть потрібну відповідь***

1. назва (ім’я), яку користувач надає об’єктам, наприклад, змінних, сталих, функціям.
2. пойменована ділянка оперативної пам’яті комп’ютера, де зберігається значення деякої величини;
3. програма, яка перекладає вихідний код мови на мову зрозумілу комп’ютеру;

**20. Математичні функції sin(), cos() належать до такої бібліотеки:**

***Оберіть потрібну відповідь***

1. math.h.
2. iostream.h;
3. conio.h;
4. Немає правильної відповіді;

**21. Що виконує фрагмент програми:**

**for( int i=0; i <5;i++)**

**{**

**a=rand()%100;**

**massiv[i]=a;**

**cout << '\t' << massiv[i];**

**}**

***Оберіть потрібну відповідь***

а) заповнює елементи масиву вибірковим чином

б) заповнює елементи масиву числами, введеними з клавіатури

в) заповнює елементи масиву константою та виводить масив на екран

г) виводить масив до файлу

**22. Яка функція С++ дозволяє об’єднати рядки?**

***Оберіть потрібну відповідь***

1. strcat(tx1,tx2);
2. charcat(tx1,tx2);
3. stringcat(tx1,tx2);
4. strplus(tx1,tx2);
5. charplus(tx1,tx2);
6. stringplus(tx1,tx2);

**23. Оберіть функцію якій відповідає математичний запис: √x.**

1. abs (x);
2. sqrt (x).
3. exp (x);

**24.** Цикл – це:

***Оберіть потрібну відповідь***

1. процес виконання складеної команди;
2. процес виконання певного набору команд деяку кількість разів:
3. процес повторення команди if;

**25.** Є такі види розгалужень:

***Оберіть потрібну відповідь***

1. повне, неповне, вибір.
2. складне, просте, неповне;
3. складні, структуровані;

**26.** Логічні вирази — це

***Оберіть потрібну відповідь***

1. фрагмент тексту програми, який слугує для пояснення;
2. вирази, що складаються з висловлювань, які можуть бути з’єднані логічними зв’язками.
3. комбінації спеціальних символів, які використовуються для введення виведення даних;
4. тип даних в комп'ютерних мовах програмування;

**27.** Програми складаються із синтаксичних конструкцій, які називаються ….

***Оберіть потрібну відповідь***

1. програмними кодами;
2. командами.
3. програмними агентами;

**28.** Який з наданих нижче операторів є оператором порівняння двох змінних?

***Оберіть потрібну відповідь***

1. ==
2. equal
3. =
4. :=

**29.** Змінні, які оголошені у тілі деякої функції або у блоці, називаються….

***Оберіть потрібну відповідь***

1. специфічними;
2. спеціальними;
3. локальними.

**30.** Що таке файл?

***Оберіть потрібну відповідь***

1. це сукупність даних, які розміщені на зовнішньому носії, зокрема на жорсткому диску:
2. форма подання послідовності символів в комп'ютері;
3. сутність, що дозволяє отримати доступ до певного ресурсу обчислювальної системи і має такі ознаки;

31. **Скільки разів можна звертатися до функції**?

***Оберіть правильну відповідь***

A) 1;

B) 7;

C) 100;

D) Безліч.

32. . **Функцію користувача оголошують так:**

***Оберіть правильну відповідь***

A) <тип функції> <назва функції> (<список формальних параметрів>);

B) <назва функції> (<список формальних параметрів>) <тип функції>;  
C) <назва функції> <тип функції> (<список формальних параметрів>);  
D) (<список формальних параметрів>) <тип функції> <назва функції>;

**ПИТАННЯ НА ЗАЛІК**

1. Історія розвитку ЕОМ та програмування як науки. Розвиток програмування в Україні.
2. Етапи розв’язування задач на ЕОМ.
3. Алгоритм, властивості алгоритму.
4. Процес алгоритмізації та його етапи.
5. Правила побудови алгоритму.
6. Поняття декомпозиції та інтеграції.
7. Типи та базові структури алгоритмів.
8. Базова структура алгоритму «слідування».
9. Базові структури алгоритмів «розгалуження» (if .. else...;) та «обхід» (if ..;).
10. Базова структура алгоритму «багатоваріантний вибір» (switch).
11. Базова структура алгоритму «цикл з передумовою» (**while…**).
12. Базова структура алгоритму «цикл з післяумовою».(do… **while**)
13. Базова структура алгоритму «цикл for»
14. Методології розробки програм.
15. Загальні принципи роботи з файлами в мові С++; операції читання та запису інформації для файлів різних типів.
16. Сутність і види інформаційних систем
17. Історія створення мови С++.
18. Складові частини інтегрованого середовища програмування; поняття вхідного, об'єктного та виконуваного кодів програми..
19. Визначення інформації та даних, зв’язок між ними.
20. Структура програми, написаної на мові С++.
21. Основні елементи мови С++.
22. Типи операторів мови С++.
23. Типи даних, що використовуються в мові С++ (у вигляді схеми).
24. Які основні та складені типи даних має мова С++
25. Що таке «область дії ідентифікатора» та «клас пам’яті»?
26. Тривалість життя об’єктів. Коли і як застосовується автоматичне, статичне і динамічне виділення пам’яті?
27. Що таке пріоритет виконання операцій? Перерахуйте відомі Вам операції мови С++ в порядку зростання старшинства?
28. Способи представлення дійсних чисел.
29. Арифметичні вирази, пріоритет виконання арифметичних операцій.
30. Логічні вирази, пріоритет виконання логічних операцій.
31. Поняття лінійної програми.
32. Процедури введення даних з клавіатури та виведення даних на екран.
33. Поняття розгалуженої програми. Дати характеристику умовному оператору IF.
34. Поняття розгалуженої програми. Дати характеристику оператору вибору CASE.
35. Поняття циклічної програми. Дати характеристику оператору циклу FOR.
36. Оператор циклу WHILE.
37. Охарактеризуйте типи циклічних операторів? В яких випадках застосовується той чи інший тип оператора?
38. Правила побудови алгоритмів
39. Види алгоритму
40. Способи записів алгоритмів
41. Що таке транслятор, компілятор, інтерпретатор?
42. Що таке директиви препроцесора? Яке їх призначення?
43. Що таке системи програмування?
44. Символьні величини, **виконання дій з елементами символьного масиву**.
45. Рядкові величини, стандартні функції для роботи з ними.
46. Поняття масиву; типи масивів; операції над масивами.
47. Поняття структури; типи структур; операції над елементами структур.
48. Як у С++ реалізується введення-виведення елементів масиву та структури?
49. Мови програмування
50. Поняття функції та підпрограми; використання локальних та глобальних змінних, формальних та фактичних параметрів.
51. Класифікація стандартних процедур та функцій мови С++.
52. Основні задачі інформаційної системи
53. Основні функції інформаційних систем та їх зміст.
54. Задачі оброблення інформації.
55. Класифікація структур даних. Базові одиниці структур даних.
56. Лінійні та нелінійні структури даних. Приклади.
57. Принципи побудови і аналізу алгоритмів.
58. Тестування програм: визначення та призначення.
59. Види тестування.
60. Поняття файлу, операції з файлами.
61. Організація роботи з файлами.
62. Поняття покажчика та адреси.