

Лабораторна робота №2

Моделі педагогічного тестування

Завдання:

- 1. Ознайомитися з основними моделями педагогічного тестування.
Визначити їх переваги і недоліки.*
- 2. Здійснити добір тестових завдань з інформатики за темою (див. особистий варіант) на ресурсі Каталог тестів.*
- 3. Визначити для яких моделей тестування відповідають відібрані завдання. Відповідь обґрунтувати.*
- 4. Побудувати алгоритми реалізації розглянутих моделей педагогічного тестування.*

Основні моделі педагогічного тестування:

1. Класична модель.

Перша й найпростіша модель тестування. Є N завдань з певного предмета, розділу чи теми або з кількох предметів. З усієї множини завдань вибирають і пропонують тестованому k завдань ($k < N$), він позначає правильні, на його думку, відповіді. Результатом тестування є відсоток правильних відповідей.

Перевага цієї моделі — її простота.

Недоліки:

- Оскільки відбір завдань — це процес випадковий, не , можна передбачити, завдання якої складності отримає тестований, тому одному дістанеться k легких, а іншому — k складних завдань.
- Оцінка залежить від кількості правильних відповідей, не враховується складність завдань. Через свої недоліки класична модель тестування має найнижчу надійність, тому що відсутність обліку параметрів завдань не дає можливості об'єктивно оцінити знання тестованого. Останнім часом

спостерігається тенденція до використання досконаліших моделей тестування, зокрема адаптивного тестування.

2. Класична модель з урахуванням складності завдань.

Є N завдань із певного предмета, розділу чи теми або з кількох предметів.

Кожне завдання має певний рівень складності.

З усієї множини завдань випадковим методом вибирають k завдань ($k < N$), які пропонують тестованому. Результат відповідей на кожне завдання позначають «правильно» або «неправильно». Підсумовуючи результати тестування, ураховують складність запитань, на які тестований відповів правильно. Чим складніше завдання, тим вищим буде результат тестування. Складність не враховують для завдань, які виконано неправильно.

Недолік: через випадковість вибірки не можна заздалегідь визначити, які завдання за складністю отримає тестований, тому одному учаснику тестування дістанеться k легких, а іншому — k складних завдань. Моделі з урахуванням складності завдань дають можливість більш адекватно оцінити знання тестованого, але випадковий вибір завдань не дає можливості підготувати паралельні за складністю тести, тобто однакової сумарної характеристики складності завдань, що зменшує надійність тестування.

3. Модель із зростанням складності.

Є N завдань із певного предмета, розділу чи теми або з кількох предметів.

Кожне завдання має певний рівень складності. Усі завдання розподілено між t рівнями складності. У тесті мають бути завдання всіх рівнів складності. З усієї сукупності випадковим методом вибирають k завдань ($k < N$) й упорядковують їх за зростанням рівня складності. Кількість завдань кожної групи складності має бути однаковим або підпорядковане нормальному закону. Результат тестування визначають аналогічно до моделі 2. Така модель забезпечує рівномірність розподілу завдань за складністю, а отже, надійність результатів тестування вища, ніж у попередніх моделях.

4. Модель з урахуванням часу виконання завдання.

У цій моделі, визначаючи результат тестування, ураховують час, за який тестований виконував кожне завдання. Це роблять для того, щоб врахувати можливість несамотійної відповіді на завдання: тестований може шукати відповідь у підручнику чи будь-якому іншому джерелі, але його оцінка буде низькою, навіть якщо завдання він виконав правильно. З іншого боку, якщо він не скористався підказкою, але довго думав над завданням, це означає, що він не достатньо добре вивчив теорію, а отже, навіть коли він відповів правильно,

оцінку йому буде знижено. Час для виконання може задавати як константа для всіх завдань тесту або розраховуватися для кожного окремого завдання з урахуванням його складності, оскільки для відповіді на складне питання потрібно часу значно більше, ніж на просте. Ця модель дає можливість підвищити надійність результатів тестування, особливо поєднавши її з урахуванням складності завдань.

5. Модель з обмеженням часу на виконання тесту.

Є N завдань з певного предмета, розділу чи теми або з кількох предметів. Із цієї множини завдань довільно вибирають k завдань ($k < N$) і вказують максимальний час для виконання тесту. Щоб оцінити результат тестування, враховують тільки ті завдання, на які тестований за відведений час відповів правильно. Сам тест може бути побудовано за моделями 1-4.

6. Адаптивна модель.

Ця модель спирається на класичну модель з урахуванням складності завдань. Вид комп'ютерного тестування, при якому тестові завдання з певними характеристиками послідовно зображуються на екрані комп'ютера, а рівень підготовки тестованого, із зростаючою точністю оцінюється відразу ж після комп'ютерної відповіді. Кожне наступне завдання в адаптивному тестуванні залежить від попередніх відповідей: кожне наступне завдання буде складнішим, якщо попереднє завдання тестований виконав правильно. Якщо у попередньому завданні була допущена помилка, наступне завдання комп'ютерна програма запропонує легше. Кількість завдань тесту заздалегідь не фіксують, а процес тестування завершують після досягнення заданої точності оцінки рівня підготовленості тестованого. Відбувається це тоді, коли тестований виходить на певний постійний рівень складності, наприклад, відповідає поспіль на певну критичну (заздалегідь визначену) кількість запитань одного рівня складності.

Переваги:

1. Дає можливість гнучко виміряти знання тестованих.
2. Дає можливість виміряти знання меншою кількістю завдань, ніж у класичній моделі.
3. Виявляє теми, які тестований знає погано і дає можливість перевірити знання з цієї теми додатково.

Недоліки:

1. Заздалегідь невідомо, скільки запитань треба поставити тестованому, щоб визначити рівень його знань.

2. Можна застосувати модель тільки за наявності комп'ютера. Надійність результатів такого тестування найвища, тому що програма тестування підлаштовується під рівень знань тестованого.

7. Модель тестування за сценарієм.

Ця модель є продовженням й удосконаленням класичної моделі тестування та намагається усунути недоліки класичної моделі тестування. Викладач перед проведенням тестування формує його сценарій, у якому враховує:

- кількість завдань з кожної теми, які мають бути в тесті;
- кількість завдань кожного рівня складності, які мають бути в тесті;
- кількість завдань кожної форми, які мають бути в тесті;
- час виконання тесту;
- інші параметри тесту на розсуд викладача.

На відміну від «Моделі із зростанням складності», яка теж враховує складність тестів, у цій моделі автор сам визначає, скільки і яких завдань з якої саме теми має бути в тесті, отже, тестування здійснюється в абсолютно однакових умовах для всіх його учасників. Порівняно з адаптивною моделлю ця модель є менш ефективною, тому що не враховує індивідуальні особливості кожного учня. Надійність результатів тестування аналогічна до моделі зі зростанням складності.

Тестові завдання з теми “Масиви”:

1. Як ім'я масиву може бути використаний будь-який

- а) пластифікатор;
- б) містифікатор;
- в) модифікатор;
- г) скарифікатор;
- д) ідентифікатор;
- е) версифікатор.

2. Звернення до елементів масиву здійснюється за допомогою вказівки імені масиву та списку індексів

- а) між похилими рисками;
- б) у круглих дужках;
- в) у фігурних дужках;
- г) між апострофами;
- д) у квадратних дужках;
- е) між знаками менше та більше.

3. Як роздільник у списку індексів багатовимірному масиву використовується

- а) знак долара;

- б) похила риска;
- в) крапка з комою;
- г) кома;
- д) точка;
- ж) двокрапка.

4. Які з наведених нижче записів є зверненнями до елементів масиву

- а) A[1,1]
- б) A[1:1]
- в) A[1;1]
- г) A[1-1]
- д) A[1,1,1]

5. Які з наведених нижче службових слів не можуть бути використані при оголошенні масиву

- а) array
- б) of
- в) integer
- г) const
- д) type
- е) var

6. Які з наведених нижче службових слів не можуть бути використані при оголошенні масиву

- а) function
- б) end
- в) begin
- г) for
- д) while
- е) of

7. Які з наведених нижче записів не є зверненнями до елементів масиву

- а) МАМА[1,1]
- б) ПАЛЬТО[1,1]
- в) КЕНГУРУ[1,1]
- г) ВАРВАРА[1,1]
- д) DOG'S[1,1]

8. Які з наведених нижче записів не є зверненнями до елементів масиву

- а) MAS[1,sqr(4)]
- б) MAS[1,sqrt(5)]
- в) MAS [1, sqrt (]
- д) MAS[sqr(4),sqr(2)]
- д) MAS[1/10,10/1]
- е) MAS[1+6,sqr(5)+1]

9. Індеси в масиві можуть бути будь-якого порядкового типу, крім типу

- а) Integer
- б) Char
- в) LongInt

10. Для двовимірного масиву вважають, що перший індекс означає номер _____, а другий - номер _____ уявної таблиці, елементами якої є відповідні значення даного масиву.

11. Наступний фрагмент коду певної програми

```
Const MaxN=10; MaxM=15; var tmyarray : array[1..MaxN,1..MaxM] of integer;
```

визначає масив із ім'ям _____.

12. Наступний фрагмент коду деякої програми

```
Const MaxN=3; MaxM=3; tmyarray : array[1..MaxN,1..MaxM] of integer = ((1,0,0),(0,1,0),(0,0,1));
```

визначає масив-константу.

Побудуйте відповідну їй уявну матрицю.

Відповідь:

13. У програмі необхідно буде працювати з масивами такого типу:

false	false	true
false	true	false

Допишіть необхідну інформацію у фрагмент коду програми, який вирішує це завдання

```
Const MaxN= _____; Var MaxM= _____; tmyarray : array[1..MaxN,1..MaxM] of _____;
```

14. У програмі визначено двовимірний масив tmyarray.

```
Const
```

```
MaxN = 2;
```

```
MaxM = 3;
```

```
Var
```

```
tmyarray :
```

```
array[1..MaxN,1..MaxM] of boolean;
```

Кількість рядків у цьому масиві дорівнює _____.

15. Кількість елементів у двовимірному масиві визначається шляхом множення числа рядків на число _____ цього масиву.

16. Кількість пам'яті для зберігання масиву визначається шляхом множення числа елементів масиву на число _____, необхідне для зберігання окремої компоненти масиву.

17. У програмі визначено двовимірний масив tmyarray.

```
Const
```

```
MaxN= 2;
```

```
MaxM= 3;
```

```
Var
```

```
tmyarray :
```

array[1..MaxN,1..MaxM] of boolean;

Кількість елементів у цьому масиві дорівнює _____.

18. У програмі визначено двовимірний масив tmyarray.

Const

MaxN= 2;

MaxM= 3;

Var

tmyarray :

array[1..MaxN,1..MaxM] of boolean;

Об'єм пам'яті, необхідний зберігання цього масиву, дорівнює _____ байт.

19. Формула визначення адреси осередку пам'яті, де зберігається i,j елемент двовимірного масиву Mas має вигляд $\text{addr}(\text{Mas}[i,j]) = \text{addr}(\text{Mas}[1,1]) + V * (M * (i-1) + j - 1)$, де M - кількість _____; V - число байт, необхідні зберігання елемента масиву.

20. Для організації доступу до всіх елементів двовимірного масиву необхідно використовувати вкладені _____.

21. Якщо Var a,b: array[1..5] of integer; c: array[1..5] of integer; то оператор b:=a+b;

а) дозволений

б) не дозволений

22. Елементи, обидва індекси яких рівні, утворюють головну _____ матриці.

23. Які значення зберігатимуться в масиві B після виконання наступних операторів var b:array[1..3,1..3] of integer; i,j: integer; begin for i:=1 to 3 do for j:=1 to 3 do b[i,j]:=i-j;

Відповідь:

24. Які значення зберігатимуться в масиві B після виконання наступних операторів var B:array[1..3,1..3] of integer; i,j: integer; begin for i:=1 to 3 do for j:=1 to 3 do begin b[i,j]:=0 if i=j then b[i,j]:=1 end;

Відповідь:

25. Заданий двовимірний масив a[1..n,1..n].

Фрагмент алгоритму

```
for i:=1 to n do
for j:=1 to n do
begin
c:=a[i,j];
a[i,j]:=a[n-i+1,j];
a[n-i+1,j]:=c
end;
```

- а) змінює порядок рядків таблиці
- б) змінює порядок стовпців
- в) змінює місцями елементи головної та побічної діагоналі
- г) змінює місцями елементи і-го рядка та побічної діагоналі
- д) нічого не змінює у таблиці

26. Заданий двовимірний масив $a[1..n, 1..n]$.

Фрагмент алгоритму

```
for i:=1 to n do
begin
c:=a[i,i];
a[i,i]:=a[i,n-i+1];
a[i,n-i+1]:=c
end;
```

- а) змінює порядок рядків таблиці
- б) змінює порядок стовпців
- в) змінює місцями елементи головної та побічної діагоналі
- г) змінює місцями елементи і-го рядка та побічної діагоналі
- д) нічого не змінює у таблиці

Автор тесту вказує, що усі відповіді оцінюються однаково: правильні – 1 бал, неправильні – 0 балів. Пропонується випадково обирати 10-15 завдань для кожного тесту.

Так як завдання обираються випадково (без чіткого зростання складності), відсутнє обмеження в часі виконання та всі завдання оцінюються однаково, тест можна віднести до **класичної моделі**.

Алгоритм реалізації класичної моделі педагогічного тестування включає наступні кроки:

1. Створення тесту: створення питань і відповідей для тестування, зважаючи на мету тесту, його складність та формат.
2. Адміністрування тесту: проведення тестування згідно зі створеним тестом, забезпечення потрібної інфраструктури для його проведення, такої як приміщення, комп'ютери або інші пристрої.

3. Оцінювання тесту: перевірка відповідей учнів і присвоєння балів за правильні відповіді.
4. Аналіз результатів тестування: оцінка результатів тестування, порівняння результатів зі стандартами та збору статистичних даних, що можуть бути використані для поліпшення майбутніх тестів.
5. Повідомлення про результати: надання учням результатів тестування, а також можливих рекомендацій для подальшого розвитку навчального процесу.

Ці кроки можуть змінюватись залежно від конкретної ситуації та мети тестування.