

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

Кафедра програмної інженерії та інформаційних технологій управління  
(назва)

**ПАКЕТ КОНТРОЛЬНИХ ЗАВДАНЬ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ З**  
**ДИСИЦПЛІНИ**

**ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ТЕСТУВАННЯ**

(назва навчальної дисципліни)

рівень вищої освіти перший (бакалаврський)  
перший (бакалаврський) / другий (магістерський)

галузь знань 12 Інформаційні технології  
(шифр і назва)

спеціальність 126 Інформаційні системи та технології  
(шифр і назва)

вид дисципліни професійна підготовка  
(загальна підготовка / професійна підготовка)

форма навчання денна  
(денна / заочна)

Укладач Орехов Сергій Валерійович, доцент  
(прізвище, посада)

Харків – 2018 рік

## ПОЯСНЮВАЛЬНА ЗАПИСКА

**Мета контрольних завдань** - перевірити ступінь сформованості у майбутніх фахівців знань щодо принципів використання сучасних методів тестування програмного забезпечення.

Контрольні завдання вимагають від студента творчого підходу, креативності, спонукають до пошукової діяльності. Вони орієнтовані на те, що слухачі повинні продемонструвати:

- Здатність застосовувати знання у практичних ситуаціях.
- Знання та розуміння предметної області та розуміння професійної діяльності.
- Здатність вчитися і оволодівати сучасними знаннями.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- Здатність аналізувати, вибирати і застосовувати методи і засоби для тестування.

Перелік контрольних питань для перевірки теоретичних знань, умінь та навичок додається. Вони складені на підставі навчальної програми професійної дисципліни «ЯКІСТЬ ПЗ та ТЕСТУВАННЯ» та робочого навчального плану підготовки фахівців освітньо-кваліфікаційного рівня „бакалавр” за спеціальністю 126 Інформаційні системи та технології у галузі знань 12 «Інформаційні технології».

Тривалість виконання контрольних завдань - 2 години

# НАВЧАЛЬНА ПРОГРАМА НОРМАТИВНОЇ ДИСЦИПЛІНИ

## **Модуль 1. Якість програмного забезпечення**

1. Аналіз метрик якості програмного забезпечення
2. Аналіз надійності програмного забезпечення
3. Побудова метричної моделі програмного забезпечення
4. Побудова моделі якості в ІТ компанії
5. Побудова моделі якості ІТ проекту
6. Використання метрик для оцінки якості програмного коду

**Література: основна [1 – 4]; додаткова [5 – 15].**

## **Модуль 2. Процес верифікації програмного забезпечення**

1. Аналіз методів верифікації програмного забезпечення
2. Аналіз методів валідації програмного забезпечення
3. Побудова циклу попередження дефектів в ІТ компанії
4. Використання метрик для верифікації програмного забезпечення

**Література: основна [1 – 4]; додаткова [5 – 15].**

## **Модуль 3. Тестування програмного забезпечення**

1. Побудова системи відстеження програмних помилок
2. Використання модульного тестування
3. Використання інтеграційного тестування
4. Використання системного тестування
5. Використання тестування навантаженням
6. Використання альфа-тестування
7. Використання бетта-тестування
8. Використання регресійного тестування

**Література: основна [1 – 4]; додаткова [5 – 15].**

## КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ ТА ВМІНЬ СТУДЕНТІВ

Рівень досягнень/Marks			Критерії оцінювання/Evaluation criteria		
Національна оцінка National grad		Бали Local grad	Оцінка за шкалою ЄКТС ECTS grad	позитивні/positiv	негативні/negativ
<u>Відмінно</u> Excelient	5	95-100	A	Глибоке знання навчального матеріалу, що містяться в основних і додаткових літературних; Вміння аналізувати явища, які вивчаються в їхньому взаємозв'язку і розвитку; Вміння проводити теоретичні розрахунки; Відповіді на запитання чіткі, лаконічні, логічно послідовні; Вміння застосовувати теоретичні положення під час розв'язання складних практичних задач.	
<u>Відмінно</u> Excelient	5	90-94	B	Глибокий рівень знань в обсязі обов'язкового матеріалу, що передбачений модулем; Вміння давати аргументовані відповіді на запитання і проводити теоретичні розрахунки; Вміння вирішувати складні практичні задачі.	Відповіді на запитання містять незначні неточності.
<u>Добре</u> Good	4	85-89	B	Глибокий рівень знань в обсязі обов'язкового матеріалу, що передбачений модулем; Вміння давати аргументовані відповіді на запитання і проводити теоретичні розрахунки; Вміння вирішувати складні практичні задачі.	Відповіді на запитання містять певні неточності.
<u>Добре</u> Good	4	75-84	C	Міцні знання матеріалу, що вивчається, та його практичного застосування; Вміння давати аргументовані відповіді на запитання і проводити теоретичні розрахунки; Вміння вирішувати практичні задачі.	Невміння використовувати теоритичні знання для вирішення складних практичних задач.
<u>Задовільно</u> Satisfactory	3	65-74	D	Знання основних фундаментальних положень матеріалу, що вивчається, та їх практичного застосування; Вміння вирішувати прості практичні задачі.	Невміння давати аргументовані відповіді на запитання; Невміння аналізувати викладений матеріал і виконувати розрахунки; Невміння вирішувати складні практичні задачі.

<u>Задовільно</u> Satisfactory	3	60-64	E	Знання основних фундаментальних положень матеріалу модуля; Вміння вирішувати найпростіші практичні задачі.	Незнання окремих (непринципових) питань з матеріалу модуля; Невміння послідовно і аргументовано висловлювати думку; Невміння застосовувати теоретичні положення при розв'язанні практичних задач.
<u>Незадовільно</u> Fail	2	35-59	FX	Додаткове вивчення матеріалу модуля може бути виконане в терміни, що передбачені навчальним планом.	Незнання основних фундаментальних положень навчального матеріалу модуля; Істотні помилки у відповідях на запитання; Невміння розв'язувати прості практичні задачі.
<u>Незадовільно</u> Fail	2	<35	F		Повна відсутність знань значної частини навчального матеріалу модуля; Істотні помилки у відповідях на запитання; Незнання основних фундаментальних положень; Невміння орієнтуватися під час розв'язання простих практичних задач.

**Білет № 1**

1. Назвіть фактори якості програмного забезпечення.
2. Розрахуйте метрику МакКейба для наступного програмного коду:

```
bool XZField::findNextMove(int &row, int &col)
{
    if(possibleMoves == NULL)
        possibleMoves = new int[rows * cols];

    ZeroMemory(possibleMoves, rows * cols * sizeof(int));

    for(int i = 0; i < rows; i++)
        for(int j = 0; j < cols; j++)
        {
            if(getElement(i, j) == EMPTY)
            {
                XZField f(*this);
                f.setElement(CROSS, i, j);
                analyseMove(&f, 0, i, j, possibleMoves);
            }
            else
            {
                possibleMoves[i * cols + j] = 0x80000000;
            }
        }

    bool result = selectMove(row, col);
    return result;
}
```

3. Дайте визначення поняттю верифікації програмного забезпечення.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ

**Білет № 2**

1. Назвіть основні критерії структурного тестування.
2. Розрахуйте метрику МакКейба для наступного програмного коду:

```
bool XZGame::checkCursorPosition(int xPos, int yPos)
{
    h = a.y - yPos;
    if(h < 0)
    {
        areaX = areaY = -1;
        return false;
    }
    l = L * h / H;
    innerBase = downBase - 2 * l;
    stepX = innerBase / 3;
    if(yPos > b.y)
    {
        if(yPos > ab2.y)
        {
            if(yPos > ab1.y)
            {
                if(yPos > a.y)
                {
                    areaY = -1; return false;
                }
                else { areaY = 2; }
            }
            else { areaY = 1; }
        }
        else{ areaY = 0; }
    }
    else{ areaY = -1; return false; }

    if((xPos > (a.x + l)) && (xPos < (d.x - l)))
    {
        areaX = floor((xPos - (a.x + l)) / stepX);
    }
    else{ areaX = -1; return false; }
    return true; }
}
```

3. Дайте визначення поняттю валідація програмного забезпечення.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ

**Білет № 3**

1. Назвіть основні ознаки модульного тестування.
2. *Розрахуйте метрику МакКейба для наступного програмного коду:*

```
public static void reshuffle(int[] pack) {  
  
    if(pack != null) {  
  
        int length = pack.length;  
  
        Random generator = new Random(new Date().getTime());  
  
        for(int i = 0; i < length; i++) {  
  
            int newPos = generator.nextInt(length);  
  
            int curCard = pack[i];  
  
            pack[i] = pack[newPos];  
  
            pack[newPos] = curCard;  
  
            if(i%(length/4) == 0) {  
  
                int pause = generator.nextInt(20);  
  
                try {  
  
                    Thread.currentThread().sleep(pause);  
  
                } catch (InterruptedException ex) {}  
  
                generator.setSeed(new Date().getTime());  
  
            } } } }  
}
```

3. Дайте опис циклу попередження помилок від компанії ІВМ.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ



**Білет № 4**

1. Назвіть основні ознаки регресійного тестування.
2. *Розрахуйте кількість програмних помилок (попереджень) для наступного програмного коду згідно метрик Холстеда:*

```
bool XZGame::checkCursorPosition(int xPos, int yPos)
{
    h = a.y - yPos;
    if(h < 0)
    {
        areaX = areaY = -1;
        return false;
    }
    l = L * h / H;
    innerBase = downBase - 2 * l;
    stepX = innerBase / 3;
    if(yPos > b.y)
    {
        if(yPos > ab2.y)
        {
            if(yPos > ab1.y)
            {
                if(yPos > a.y)
                { areaY = -1; return false; }
                else { areaY = 2; }
            }
            else { areaY = 1; }
        }
        else { areaY = 0; }
    }
    else { areaY = -1; return false; }

    if((xPos > (a.x + l)) && (xPos < (d.x - l)))
    {
        areaX = floor((xPos - (a.x + l)) / stepX);
    }
    else { areaX = -1; return false; }
    return true; }
}
```

3. Дайте опис звіту про програмну помилку.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ

**Білет № 5**

1. Назвіть основні метрики Холстеда.
2. Розрахуйте метрику МакКейба для наступного програмного коду:

```
public int compare(Object o1, Object o2) {  
  
    if(o1 != null && o2 != null &&  
        o1 instanceof File && o2 instanceof File) {  
  
        File f1 = (File)o1;  
        File f2 = (File)o2;  
  
        String fullPath1 = f1.getAbsolutePath();  
        String fullPath2 = f2.getAbsolutePath();  
  
        if(fullPath1.equals(fullPath2)) {  
  
            return 0;  
        }  
  
        String[] res1 = p.split(fullPath1);  
        String[] res2 = p.split(fullPath2);  
        if(res1.length > res2.length) {  
  
            return 1;  
        }  
        if(res1.length < res2.length) {  
  
            return -1;  
        }  
        if(res1.length == res2.length) {  
  
            return collator.compare(fullPath1, fullPath2);  
        }  
    }  
  
    return 0;  
}
```

3. Дайте визначення моделі комплексного управління якістю згідно стандарту ISO.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ

С.В. ОРЄХОВ

**Білет № 6**

1. Назвіть основні ролі в процесі верифікації програмного забезпечення.
2. Розрахуйте метрику МакКейба для наступного програмного коду:

```
int main()
{
const int NX = 15;
const int NY = 15;
int Nums[NX][NY];

for (int i = 0; i < NX; i++)
    for (int j = 0; j < NY; j++) Nums[i][j] = 0;
randomize();
int a, b;
for (int i = 0; i < 1000000; i++)
    { a = random(NX); b = random(NY); Nums[a][b]++; }
for (int i = 0; i < NX; i++)
    {
    for (int j = 0; j < NY; j++)
        { std::cout.width(5); std::cout << Nums[i][j]; }
    std::cout << '\n';
    }
getch();

clrscr();
int f[25];
for (int i = 0; i < 25; i++)
    f[i] = 0;
for (int i = 0; i < NX; i++)
    for (int j = 0; j < NY; j++)
        { int c = Nums[i][j] - 4194;
        if (c >= 0 && c < 500) f[c / 20]++;
        }
for (int i = 0; i < 25; i++)
    {
    for (int j = 0; j < f[i]; j++) std::cout << '-';
    std::cout << '\n';
    }
getch();
}
```

3. Дайте визначення стохастичному критерію тестування програмного забезпечення.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ

**Білет № 7**

1. Назвіть концепцію тестування програмного забезпечення.
2. *Розрахуйте метрику МакКейба для наступного програмного коду:*

```
bool BaseWindowClass::Create(HINSTANCE hInstance, LPSTR szTitle,
                             DWORD iStyle, int iWidth, int iHeight)
{
    baseWndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
    baseWndClass.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_ICON1));
    baseWndClass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
    baseWndClass.hInstance = hInstance;
    baseWndClass.lpfWndProc = WndProc;
    baseWndClass.lpszClassName = "Base Window";
    baseWndClass.lpszMenuName = NULL;
    baseWndClass.cbClsExtra = 0;
    baseWndClass.cbWndExtra = 0;
    baseWndClass.style = 0;

    if(!RegisterClass(&baseWndClass))
    {ExceptionBase e(CLASSREGISTRATIONERROR, "Не могу зарегистрировать класс");
      throw e;}

    baseHWindow = CreateWindow("Base Window", szTitle, iStyle,
                              CW_USEDEFAULT, CW_USEDEFAULT,
                              iWidth, iHeight, NULL, NULL,
                              hInstance, (void*)this);

    if(baseHWindow == NULL)
    {ExceptionBase e(WINDOWCREATIONERROR, "Не могу создать окно");
      throw e;}

    if(!SetWindowText(baseHWindow, szTitle))
    {ExceptionBase e(SETTITLEERROR, "Не могу изменить текст в заголовке окна");
      throw e;
    }
    return 0;}

```

3. Дайте визначення поняття «клас еквівалентності».

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ

С.В. ОРЄХОВ

**Білет № 8**

1. Назвіть основні етапи процесу тестування.
2. *Розрахуйте метрику МакКейба для наступного програмного коду:*

```
void NewGameMenu::createVertexBuffer()
{
    if(FAILED(pD3DDevice->CreateVertexBuffer(4*sizeof(XZGAMEMENUVERTEX),
        0, D3DFVF_XZGAMEMENUVERTEX, D3DPOOL_DEFAULT, &pVertices, NULL)))
    { ExceptionBase err(VERTEXBUFFERCREATIONERROR, "Не могу создать Vertex Buffer");
      throw err;
    }

    XZGAMEMENUVERTEX *pVer;
    if(FAILED(pVertices->Lock(0, 0, (void**)&pVer, 0)))
    { ExceptionBase err(VERTEXBUFFERLOCKERROR, "Ошибка при вызове метода Lock");
      throw err;
    }
    pVer[0].position = D3DXVECTOR3(-0.5f, 0.0f, -0.5f);
    pVer[0].vecNormal = D3DXVECTOR3(0.0f, 1.0f, 0.0f);
    pVer[0].tu = 0.0f;
    pVer[0].tv = 1.0f;
    pVer[1].position = D3DXVECTOR3(-0.5f, 0.0f, 0.5f);
    pVer[1].vecNormal = D3DXVECTOR3(0.0f, 1.0f, 0.0f);
    pVer[1].tu = 0.0f;
    pVer[1].tv = 0.0f;
    pVer[2].position = D3DXVECTOR3(0.5f, 0.0f, -0.5f);
    pVer[2].vecNormal = D3DXVECTOR3(0.0f, 1.0f, 0.0f);
    pVer[2].tu = 1.0f;
    pVer[2].tv = 1.0f;
    pVer[3].position = D3DXVECTOR3(0.5f, 0.0f, 0.5f);
    pVer[3].vecNormal = D3DXVECTOR3(0.0f, 1.0f, 0.0f);
    pVer[3].tu = 1.0f;
    pVer[3].tv = 0.0f;
    pVertices->Unlock();
}
```

3. Дайте класифікацію програмних помилок.

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ

С.В. ОРЄХОВ

Білет № 9

1. Назвіть основну проблему тестування програмного забезпечення.
2. *Розрахуйте кількість помилок (попереджень) для наступного програмного коду згідно метрики Холстеда:*

```
bool XZGame::checkCursorPosition(int xPos, int yPos)
{
    h = a.y - yPos;
    if(h < 0)
    {
        areaX = areaY = -1;
        return false;
    }
    l = L * h / H;
    innerBase = downBase - 2 * l;
    stepX = innerBase / 3;
    if(yPos > b.y)
    {
        if(yPos > ab2.y)
        {
            if(yPos > ab1.y)
            {
                if(yPos > a.y)
                {
                    areaY = -1; return false;
                } else { areaY = 2; }
            } else { areaY = 1; }
        } else { areaY = 0; }
    } else { areaY = -1; return false; }

    if((xPos > (a.x + l)) && (xPos < (d.x - l)))
    {
        areaX = floor((xPos - (a.x + l)) / stepX);
    }
    else { areaX = -1; return false; }
    return true; }
}
```

3. Дайте визначення поняттю «доброго критерію тестування».

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЕХОВ

**Білет № 10**

1. Назвіть основні критерії функціонального тестування.
2. Розрахуйте прогнозне значення кількості програмних помилок для наступного програмного коду за метрикою Холстеда:

```
void XZMesh::loadMesh(const char *fileName)
{
    if(fileName == NULL)
    {
        ExceptionBase err(MESHNOTSPECIFIEDERROR, "Имя файла не указано");
        throw err;
    }
    LPD3DXBUFFER matBuffer;
    if(FAILED(D3DXLoadMeshFromX(fileName, D3DXMESH_SYSTEMMEM,
pD3DDevice, NULL, &matBuffer, NULL, &matNumber, &pMesh)))
    {
        char *mes = "Îá ïñáó çàãðóçèðü mesh: ";
        size_t mesLength = strlen(mes) + strlen(fileName);
        char *fullMes = new char[mesLength + 1];
        strcpy(fullMes, mes);
        strcat(fullMes, fileName);
        ExceptionBase err(MESHNOTFOUNDERERROR, fullMes);
        delete [] fullMes;
        throw err;
    }

    D3DXMATERIAL *d3dxMat = (D3DXMATERIAL*)matBuffer->GetBufferPointer();
    pMaterials = new D3DMATERIAL9[matNumber];

    for(int i = 0; i < matNumber; i++)
    {
        pMaterials[i] = d3dxMat[i].MatD3D;
        pMaterials[i].Ambient = pMaterials[i].Diffuse;
    }

    matBuffer->Release();
}
```

3. Дайте визначення поняттю «добрий тест».

Затверджено на засіданні кафедри інформаційних систем протокол № 09 від 20.02.2016 р.

Завідувач кафедри  
професор  
Екзаменатор  
доцент

М.Д. ГОДЛЕВСЬКИЙ  
С.В. ОРЄХОВ

# НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Орехов С.В. Лабораторний практикум з дисципліни “Якість ПЗ та тестування”  
[Електронний ресурс]. – Режим доступу: ntumoodle.com

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### *Базова*

1. Канер Сэм. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений.: Пер. с англ. Сэм Канер, Джек Фолк, Енг Кек Нгуен. — К.: Издательство «Диасофт», 2001.
2. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер; - Питер, 2004.
3. Сеницын С. В., Налютин Н. Ю. Верификация программного обеспечения. Учебное пособие [текст] / С. В. Сеницын, Н. Ю. Налютин; - Бином, 2008.
4. Коликова Т.В., Котляров В.П. Основы тестирования программного обеспечения. Учебное пособие [текст] / Т.В. . Коликова, В.П. Котляров; - Интуит, 2006.

### *Допоміжна література*

5. Брауде Э.Д. Технология разработки программного обеспечения / Э.Д. Брауде; - Питер, 2004.
6. Винниченко И. В. Автоматизация процессов тестирования. Питер, 2005.
7. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование – Вильямс, 2002.
8. Касперски К. Техника отладки программ без исходных текстов – СПб, БХВ - 2005.
9. Макгрегор Д., Сайкс Д. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие - ТИД «ДС», 2004.
10. Плаксин М. Тестирование и отладка программ - для профессионалов будущих и настоящих - Бином. Лаборатория знаний, 2007.
11. Роберт М., Ньюкирк Д., Быстрая разработка программ: принципы, примеры, практика - Вильямс, 2004.
12. Рэшка Д., Пол Д., Дастин Э. Автоматизированное тестирование программного обеспечения - Лори, 2003.
13. Спольски Д. Лучшие примеры разработки ПО - Питер, 2007.
14. Тамре Л. Введение в тестирование программного обеспечения [текст] / Л. Тамре; - Вильямс, 2003.
15. Фолк Д., Нгуен Е. К., Канер С. Тестирование программного обеспечения - Диасофт, 2003.