

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

Кафедра програмної інженерії та інформаційних технологій управління
(назва)

ПАКЕТ КОМПЛЕКСНИХ КОНТРОЛЬНИХ РОБІТ (ККР) ДЛЯ
ПЕРЕВІРКИ ЗНАНЬ З ДИСЦИПЛІНИ

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 тестування ПЗ.

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 ПЗ

Тема 1. Поняття якості програмного забезпечення. Поняття надійності програмного забезпечення. Модель комплексного управління якістю ISO. Модель управління якістю процесу розробки програмного забезпечення CMMI.

Тема 2. Модель оцінки якості програми ISO. Основи метричної теорії оцінки якості програми. Огляд метрик оцінки якості на рівні функціонального програмування.

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

Модуль 2. Процес верифікації ПЗ

Тема 3. Огляд моделей життєвого циклу розробки програмного забезпечення. Цикл попередження дефектів компанії IBM. Вартість усунення дефектів програмного забезпечення на різних стадіях життєвого циклу розробки. Задачі валідації та верифікації, які вирішуються на різних етапах життєвого циклу. Перелік звітної документації.

Тема 4. Визначення, ролі, обов'язки, етапи. Огляд методів верифікації. Метрики процесу верифікації. Документація про процес верифікації.

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

Модуль 3. Тестування ПЗ

Тема 5. Проблеми тестування. Фази тестування. Критерії вибору тестів. Типи тестування. Рівні тестування. Методи тестування. Модульне тестування. Інтеграційне тестування. Функціональне тестування. Тестування „чорного ящика”. Компонентне тестування. Системне тестування.

Тема 6. Критерії та класифікація програмних помилок. Аналіз помилки, що повторюється. Звіт про програмну помилку. Системи відстеження програмних помилок.

Тема 7. Характеристики доброго тесту. Класи еквівалентності та граничні умови. Тестування переходів між станами. Тестовий план. Тестовий приклад. Виконання тестів.

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.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення
122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 1.

1. Концепція тестування програмного забезпечення.
2. Провести розрахунок метрики МакКайба 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. Фактори якості програмного забезпечення.

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення

122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 2.

1. Функціональні критерії тестування.
2. Згідно метрики Холстеда треба встановити перелік програмних помилок (застережень) 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 бетта тестуванню.

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення
122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 3.

1. Основні проблеми тестування програмного забезпечення.
2. Згідно метрики Холстеда треба розрахувати оцінку кількості помилок 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. Що таке альфа тестування?

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення
122 Комп'ютерна наука

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 4.

1. Стохастичний критерій тестування.
2. Згідно метрики Холстеда треба розрахувати оцінку кількості помилок для наступного програмного коду:

```
public void actionPerformed(ActionEvent e)
{
    try
    { String src = expressionField.getText(); converter = new PostfixConverter(src);
      calc = new PostfixCalculator(converter.convertToPostfix());
      result = new ComplexNumber(calc.calculate()); String res = result.toString();
      resultField.setText(res); listModel.addElement(src + " = " + res);
      int lastElement = listModel.size() - 1; historyList.setSelectedIndex(lastElement);
      historyList.ensureIndexIsVisible(lastElement); }
    catch(IncorrectElementException iee) {
      listModel.addElement(iee.getMessage()); int lastElement = listModel.size() - 1;
      historyList.setSelectedIndex(lastElement); historyList.ensureIndexIsVisible(lastElement);
      expressionField.select(iee.getIndex(), expressionField.getText().length()); }
    catch(Exception err) {
      listModel.addElement(err.getMessage()); int lastElement = listModel.size() - 1;
      historyList.setSelectedIndex(lastElement); historyList.ensureIndexIsVisible(lastElement); }
}
```

3. Що таке альфа тестування?

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення
122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 5.

1. Методика аналізу помилки, що повторюється.
2. Згідно метрики МакКейба треба розрахувати оцінку кількості тестів для наступного програмного коду:

```
public void actionPerformed(ActionEvent e)
{
    try
    { String src = expressionField.getText(); converter = new PostfixConverter(src);
      calc = new PostfixCalculator(converter.convertToPostfix());
      result = new ComplexNumber(calc.calculate()); String res = result.toString();
      resultField.setText(res); listModel.addElement(src + " = " + res);
      int lastElement = listModel.size() - 1; historyList.setSelectedIndex(lastElement);
      historyList.ensureIndexIsVisible(lastElement); }
    catch(IncorrectElementException iee) {
      listModel.addElement(iee.getMessage()); int lastElement = listModel.size() - 1;
      historyList.setSelectedIndex(lastElement); historyList.ensureIndexIsVisible(lastElement);
      expressionField.select(iee.getIndex(), expressionField.getText().length()); }
    catch(Exception err) {
      listModel.addElement(err.getMessage()); int lastElement = listModel.size() - 1;
      historyList.setSelectedIndex(lastElement); historyList.ensureIndexIsVisible(lastElement); }
}
```

3. Що таке тестування «білий ящик»?

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення
122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 6.

1. Цикл попередження появи дефектів.
2. Згідно метрики МакКейба треба розрахувати оцінку кількості тестів для наступного програмного коду:

```
public void init() {  
    try {  
        Dimension dim = getSize(); w = dim.width; h = dim.height; setColors(); int radius = 0;  
        if(w < h) radius = w/2;  
        else radius = h/2;  
        mNeedle = new Needle(radius, radius/5);  
        hNeedle = new Needle(2d/3d*radius, radius/4);  
        sNeedle = new Needle(radius, radius/10);  
        dial = new ClockDial(radius*2, dialColor, hairLineColor);  
        bi = (BufferedImage)createImage(w, h);  
        big = bi.createGraphics();  
        big.setRenderingHint(RenderingHints.KEY_ANTIALIASING,  
                             RenderingHints.VALUE_ANTIALIAS_ON);  
        timer = new Thread(this);  
        timer.start();  
    }  
    catch(Exception error) { System.out.println(error.getMessage()); }  
}
```

3. Що таке тестування «чорний ящик»?

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення

122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 7.

1. Цикл попередження появи дефектів.
2. Згідно метрики МакКейба треба розрахувати оцінку кількості тестів для наступного програмного коду:

```
public ClockDial(double diam, Color dialColor, Color hairLineColor) throws
IncorrectParametersException {
    if(diam <= 0) throw new IncorrectParametersException();
    this.diam = diam;    this.dialColor = dialColor;    this.hairLineColor = hairLineColor;
    dial = new BufferedImage((int)diam, (int)diam, BufferedImage.TYPE_INT_ARGB);
    Rectangle2D.Double r = new Rectangle2D.Double(-diam/40, -diam/60, diam/20, diam/30);
    AffineTransform at = new AffineTransform();    at.translate(diam/2, diam/2);    AffineTransform
prev = (AffineTransform)at.clone();    Graphics2D g2 = dial.createGraphics();
g2.setBackground(dialColor);    g2.clearRect(0, 0, (int)diam, (int)diam);    g2.setColor(hairLineColor);
    for(int i = 0; i < 12; i++)
    {    at.translate(-(diam/2 - r.width), 0);        at.rotate(Math.toRadians(30*i), diam/2 - r.width, 0);
        g2.setTransform(at);        g2.fill(r);        at.setTransform(prev);    }    }
```

3. Призначення та основні компоненти звіту про помилку

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20__ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення

122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 8.

1. Класифікація програмних помилок.
2. Згідно метрики Холстеда встановити перелік програмних помилок (застережень) для наступного програмного коду:

```
public ClockDial(double diam, Color dialColor, Color hairLineColor) throws
IncorrectParametersException {
    if(diam <= 0) throw new IncorrectParametersException();
    this.diam = diam;    this.dialColor = dialColor;    this.hairLineColor = hairLineColor;
    dial = new BufferedImage((int)diam, (int)diam, BufferedImage.TYPE_INT_ARGB);
    Rectangle2D.Double r = new Rectangle2D.Double(-diam/40, -diam/60, diam/20, diam/30);
    AffineTransform at = new AffineTransform();    at.translate(diam/2, diam/2);    AffineTransform
prev = (AffineTransform)at.clone();    Graphics2D g2 = dial.createGraphics();
g2.setBackground(dialColor);    g2.clearRect(0, 0, (int)diam, (int)diam);    g2.setColor(hairLineColor);
    for(int i = 0; i < 12; i++)
    {    at.translate(-(diam/2 - r.width), 0);        at.rotate(Math.toRadians(30*i), diam/2 - r.width, 0);
        g2.setTransform(at);        g2.fill(r);        at.setTransform(prev);    }    }
```

3. Фактори якості програмного забезпечення

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення

122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 9.

1. Модульне тестування.

2. Згідно метрики МакКейба треба розрахувати оцінку кількості тестів для наступного програмного коду:

```
void XZField::analyseMove(XZField *f, int recursionDepth, int startRow, int startCol, int *posMoves)
{
    if(recursionDepth >= recursionDepth) return;
    if(f->isGameOver(w))
    {
        if(recursionDepth % 2 == 0) { posMoves[startRow * cols + startCol] +=
            (int)pow(10, recursionDepth - recursionDepth); }
        Else { posMoves[startRow * cols + startCol] +=
            (int)((-1) * pow(10, recursionDepth - recursionDepth)); return; }
        recursionDepth++;
        for(int i = 0; i < rows; i++) for(int j = 0; j < cols; j++) {
            if(f->getElement(i, j) == EMPTY) { XZField tempField(*f);
                if(recursionDepth % 2 == 0) { tempField.setElement(CROSS, i, j);
                    analyseMove(&tempField, recursionDepth, startRow, startCol,
                        posMoves); }
                else { tempField.setElement(ZERO, i, j);
                    analyseMove(&tempField, recursionDepth, startRow, startCol,
                        posMoves); } } } }
```

3. Фактори якості програмного забезпечення

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

рівень вищої освіти перший (бакалаврський)

галузь знань 12 Інформаційні технології

спеціальність 121 Інженерія програмного забезпечення

122 Комп'ютерна науки

Навчальна дисципліна **ЯКІСТЬ ПЗ та ТЕСТУВАННЯ**

КОМПЛЕКСНА КОНТРОЛЬНА РОБОТА № 10.

1. Структурні критерії модульного тестування.
2. Згідно метрики МакКейба треба розрахувати оцінку кількості тестів для наступного програмного коду:

```
void XZField::analyseMove(XZField *f, int recNumber, int startRow, int startCol, int *posMoves)
{
    if(recNumber >= recursionDepth) return;
    if(f->isGameOver(w))
    {
        if(recNumber % 2 == 0) { posMoves[startRow * cols + startCol] +=
                                (int)pow(10, recursionDepth - recNumber); }
        Else { posMoves[startRow * cols + startCol] +=
              (int)((-1) * pow(10, recursionDepth - recNumber)); } return;
    }
    recNumber++;
    for(int i = 0; i < rows; i++) for(int j = 0; j < cols; j++) {
        if(f->getElement(i, j) == EMPTY) { XZField tempField(*f);
            if(recNumber % 2 == 0) { tempField.setElement(CROSS, i, j);
                analyseMove(&tempField, recNumber, startRow, startCol,
                    posMoves);
            }
            else { tempField.setElement(ZERO, i, j);
                analyseMove(&tempField, recNumber, startRow, startCol,
                    posMoves); } } } }
```

3. Основні проблеми тестування програмного забезпечення

Затверджено на засіданні Кафедри програмної інженерії та інформаційних технологій управління

Протокол № _____ від „_____” _____ 20____ року

Завідувач кафедри _____ Годлевський М.Д.

Укладач _____ Орехов С.В.

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 – Вільямс, 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.