**1.Розміщення** учасників тренінгу у гостинному дворі “Кільце Карпат” здійснюється в двох корпусах: триповерховому основному корпусі (24 особи) та в двоповерховому корпусі “В долині” (10 осіб), віддаль між якими становить близько 300 м по трасі та близько 100 м прямою стежкою через подвір’я, згідно схеми:



Вартість проживання в усіх двомісних номерах основного корпусу становить 500 грн з особи на добу (1000 грн за номер), в чотиримісних номерах основного корпусу - 350 грн з особи на добу (1400 грн за номер). Основний корпус має три входи: центральний – до шести номерів, по два номери на кожному поверсі (позначеними на схемі як 1.1…3.2), та два окремі входи до двоповерхових номерів підвищеної комфортності (позначеними на схемі як Л.1 та Л.2)

Вартість проживання в двомісних номерах корпусу “В долині” становить 450 грн з особи на добу (900 грн за номер), в тримісних номерах корпусу “В долині” - 300 грн з особи на добу (900 грн за номер). На першому поверсі корпусу “В долині” є кухня із всіма елементами, необхідними для приготування їжі – електроплита, посуд, холодильник. Біля корпусу “В долині” на подвір’ї знаходиться комин.

**2.Харчування** учасників здійснюється тричі на добу (в день заїзду – обід та вечеря, в день від’їзду - сніданок) в кафе біля основного корпусу. Вартість харчування – 700 грн на добу.

Учасники, які проживатимуть в корпусі “В долині”, можуть харчуватись самостійно, завдяки наявності кухні з оплатою, таким чином, тільки вартості проживання. Для учасників, які проживатимуть в основному корпусі, самостійне харчування є, певною мірою, ускладненим та нерекомендованим.

На віддалі 200 м від обидвох корпусів знаходиться продуктовий магазин з набором продуктів, достатнім для харчування. Можливим є замовлення домашніх молокопродуктів та меду (!!!) у господарів бази та місцевих мешканців.

**3.Навчання** здійснюватиметься за двома навчальними програмами, розрахованими на учнів 5-7-х класів (курс JavaForTeens) та на учнів 8-11-х класів (курс JavaBasic), що додаються нижче.

По кожному з курсів щоденно відбуватимуться два заняття тривалістю 80 хвилин – лекційне із поданням 2-3-х тем навчальної програми та практичне із розглядом кодів програмних домашніх завдань. Домашні завдання вчителі можуть виконувати самостійно (це бажано - в такому випадку коди вчителів на практичних заняттях будуть розглянутими обов’язково). Як альтернатива - спостерігання процесу написання домашніх кодів тренером.

Вчителі можуть приймати участь або в одному з обраних навчальних курсів, або зразу в обидвох. Прийняти рішення щодо обрання навчального курсу можна вже безпосередньо в Карпатах.

По кожному з навчальних курсів надається сертифікат про підвищення кваліфікації обсягом 30 годин від партнера – ТОВ “Інновейшн Девелопмент Фаундейшн”.

Вартість навчання по кожному з курсів становить 890 грн. Оплату одного або обидвох курсів можна здійснити вже в Карпатах.

Навчальний клас із медіа-засобами знаходиться біля корпусу “В долині”.

**4.Трансфер** до бази у с.Верхній Ясенів доцільно здійснити рейсовим автобусом Івано-Франківськ – Верховина, що курсує через м.Коломия і зупиняється в 100 м від бази (рейсові автобуси Івано-Франківськ – Верховина, що курсують через Яремче-Ворохту, не проходять через Верхній Ясенів). Тренер Іщеряков Сергій планує особисто зустріти всіх учасників тренінгу на залізничному вокзалі Івано-Франківська після отримання інформації про час прибуття та скоординувати подальший трансфер до Верхнього Ясенова.

**5.Оплату** проживання за 6 діб необхідно здійснити попередньо шляхом переведення коштів на картку ФОП – власника бази. Враховуючи обмежену кількість місць на базі - 34, просимо перед оплатою ознайомитись із наявністю вільних місць і зарезервувати обране місце. Проте, реальною гарантією є не резервування місця, а його оплата.

Оплату харчування в розмірі 1200 грн (28% від повної суми 4200 грн) необхідно також здійснити попередньо шляхом переведення коштів на картку ФОП – власника бази. Решту суми можна оплатити безпосередньо на базі готівкою або переведенням коштів на картку.

Оплату навчання у розмірі 890 грн за один навчальний курс необхідно здійснити на сайті idf.ua після прибуття на базу.

Оплата можливих екскурсій здійснюється безпосередньо на базі після визначення екскурсійних місць.

6.Перебування на базі є можливим разом із членами сімей.

**ДОДАТОК 1**

**Навчальна програма курсу JavaForTeens**

| **№** | **Теми навчальної програми** |
| --- | --- |
| 1 | Встановлення Eclipse. Перша програма. Поняття проєкта, пакета, класа. Виведення пустого фрейму. |
| 2 | Малювання прямокутника, кола. Робота із вісями координат комп’ютерних фреймів. АРІ пакету java.awt. Малювання автомобіля. |
| 3 | Необхідність використання змінних. Оголошення примітивних змінних. |
| 4 | Цикли. Малювання декількох автомобілів. Малювання будинку з вікнами. |
|  5 | Вкладені цикли. Малювання багатоповерхових будинків з вікнами. |
| 6 | Робота з кольорами. Малювання державних прапорів. Робота з текстом (назви держав). Почергове малювання прапорів країн з горизонтальною та вертикальною орієнтацією кольорових смуг. |
| 7 | Масиви (назви держав, кольори одного прапору). |
| 8 | Двомірні масиви (кольори прапорів різних країн). Одночасне малювання прапорів різних країн. |
| 9 | Робота з кнопками. Опрацювання подій. Зміна при натисканні кнопки: тексту, цілих чисел (інкрементер-декрементер). |
| 10 | Оператор розгалуження if-else. Змінні типу boolean. Інкрементер-декрементер із завданням режиму роботи. |

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 1.**

**Встановлення Eclipse. Перша програма. Поняття проєкта, пакета, класа. Виведення пустого фрейму.**

**Відео:** 1\_1\_firstProgram,

 1\_2\_emptyFrame.

**Домашні завдання.**

1.Встановити Java та Eclipse. Для дистанційного усунення ментором можливих проблем при встановленні Java та Eclipse рекомендовано також встановити TeamViewer.

2.Набирати 1-5 разів слова:

extends, Frame, paint, void, import, java, class, public, static, main, String, set, size, visible, true, false, super, name, title, event, window, listener, add, new, adapter, closing, system, exit, graphics.

3.Створити декілька проєктів з назвами myProject2, myProject3, …, класами motherNamePrinter, fatherNamePrinter, …, всередині класів – із методом main, в якому забезпечити виведення імен мами, тата, …

4.Створити проєкт для створення пустого фрейму з написом у лівій верхній частині фрейму, що відповідає імені розробника.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 2.**

**Виведення прямокутника, кола. Робота із вісями координат комп’ютерних фреймів. АРІ пакету java.awt. Виведення автомобіля.**

**Відео:** 2\_саrPainter.

**Код з відео:**

///////// CODE FROM VIDEO “2\_carPainter” FOR LESSON 2 /////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** AutoPainter **extends** Frame {

//////////////////// CONSTRUCTOR //////////////////////////

 AutoPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(1500, 300);

 setVisible(**true**);

 }

////////////// METHOD MAIN FOR CODE RUNNING ////////////////

 **public** **static** **void** main(String[] args) {

AutoPainter painter = **new** AutoPainter("AutoPainter");

 }

//////////// METHOD PAINT FOR SHAPES PAINTING ////////////////

 **public** **void** paint(Graphics painter) { // API

 painter.drawRect(100, 40, 150, 30); // cabin

 painter.drawRect(80, 70, 190, 40); // engine

 painter.drawOval(90, 110, 30, 30); // leftWheel

 painter.drawOval(230, 110, 30, 30); // rightWheel

 }

}

**Результат:**

 ****

**Домашні завдання.**

1. Виконати 8 завдань за посиланням

 https://learningapps.org/display?v=ph47xf6ct23

2.Написати java-код для виведення:

- будинку з дверима та вікном,

- сніговика.

 

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 3.**

**Необхідність використання змінних. Оголошення примітивних змінних.**

**Відео:** 3\_variables.

**Код з відео:**

////////// CODE FROM VIDEO “3\_variables” FOR LESSON 3 //////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** AutoPainter **extends** Frame {

///////////////////// CONSTRUCTOR //////////////////////////

 AutoPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(500, 300);

 setVisible(**true**);

 }

/////////////// METHOD MAIN FOR CODE RUNNING ////////////////

 **public** **static** **void** main(String[] args) {

 AutoPainter painter = **new** AutoPainter("AutoPainter");

 }

 ////// variables for initialization ////////////

 **int** yCabin = 40, cabinHeight = 30;

 **int** xEngine = 80, engineWidth = 190, engineHeight = 40;

 **int** diameter = 30;

 **int** shiftEngineCabin = 20, shiftEngineWheels = 10;

 ////// variables for calculation //////////////////

 **int** xCabin = xEngine + shiftEngineCabin;

 **int** cabinWidth = engineWidth - shiftEngineCabin \* 2;

 **int** yEngine = yCabin + cabinHeight;

 **int** xLeftWheel = xEngine + shiftEngineWheels;

 **int** yWheels = yEngine + engineHeight;

 **int** xRightWheel = xEngine + engineWidth - shiftEngineWheels –

diameter;

/////// METHOD PAINT FOR SHAPES PAINTING //////////////////////

 **public** **void** paint(Graphics painter) { // API

 painter.drawRect(xCabin, yCabin, cabinWidth, cabinHeight); // cabin

 painter.drawRect(xEngine, yEngine, engineWidth, engineHeight);

 // engine

 painter.drawOval(xLeftWheel, yWheels, diameter, diameter);

 // leftWheel

 painter.drawOval(xRightWheel, yWheels, diameter, diameter);

 // rightWheel

 }

}

**Домашнє завдання.** Написати java-код для виведення будинку з дверима та вікном, а також сніговика (попереднє завдання) із використанням змінних.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 4.**

**Цикли. Малювання декількох автомобілів.**

**Відео:** 4\_loops.

**Код з відео:**

////////// CODE FROM VIDEO “4\_loops” FOR LESSON 4 //////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** AutoPainter **extends** Frame {

///////////////////// CONSTRUCTOR //////////////////////////

 AutoPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(1500, 300);

 setVisible(**true**);

 }

/////////////// METHOD MAIN FOR CODE RUNNING ////////////////

 **public** **static** **void** main(String[] args) {

 AutoPainter painter = **new** AutoPainter("AutoPainter");

 }

 ////// variables for initialization ////////////

 **int** yCabin = 40, cabinHeight = 30;

 **int** xEngine = 80, engineWidth = 190, engineHeight = 40;

 **int** diameter = 30;

 **int** shiftEngineCabin = 20, shiftEngineWheels = 10;

 **int** distance = 40;

 **int** autosCount = 5;

 ////// variables for calculation //////////////////

 **int** xCabin = xEngine + shiftEngineCabin;

 **int** cabinWidth = engineWidth - shiftEngineCabin \* 2;

 **int** yEngine = yCabin + cabinHeight;

 **int** xLeftWheel = xEngine + shiftEngineWheels;

 **int** yWheels = yEngine + engineHeight;

 **int** xRightWheel = xEngine + engineWidth - shiftEngineWheels - diameter;

 **int** autosShift = engineWidth + distance;

 **public** **void** paint(Graphics painter) {

 **for**(**int** shift = 0; shift < autosCount; shift++) { // increment

 painter.drawRect(xCabin + autosShift \* shift, yCabin,

 cabinWidth, cabinHeight);

 painter.drawRect(xEngine + autosShift \* shift, yEngine,

 engineWidth, engineHeight);

 painter.drawOval(xLeftWheel + autosShift \* shift, yWheels,

 diameter, diameter);

 painter.drawOval(xRightWheel + autosShift \* shift,

yWheels, diameter, diameter);

 }

 }

}

**Результат:**



**Домашнє завдання.** Написати java-код із використанням оператора циклу for для виведення:

- довільної кількості товарних вагонів (при виведенні коліс та конекторів у вагонах НЕ ПОТРІБНО використовувати оператор for)



- пасажирського вагону із 9 вікнами.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 5.**

**Вкладені цикли. Малювання багатоповерхових будинків з вікнами.**

**Відео:** 5\_nestedLoops.

**Код з відео:**

///////// CODE FROM VIDEO “5\_nestedLoops” FOR LESSON 5 /////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** AutoPainter **extends** Frame {

 AutoPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(frameLength, frameHeight);

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) {

 AutoPainter painter = **new** AutoPainter("AutoPainter");

 } ////// variables for initialization ////////////

 **int** frameLength = 1300, frameHeight = 900;

 **int** yTopLine = 40, shiftLineAuto = 20;

 **int** cabinHeight = 30;

 **int** xEngine = 80, engineWidth = 190, engineHeight = 40;

 **int** diameter = 30;

 **int** shiftEngineCabin = 20, shiftEngineWheels = 10;

 **int** distance = 40;

 **int** autosCount = 5, laysCount = 3;

 ////// variables for calculation //////////////////

 **int** xCabin = xEngine + shiftEngineCabin;

 **int** yCabin = yTopLine + shiftLineAuto;

 **int** cabinWidth = engineWidth - shiftEngineCabin \* 2;

 **int** yEngine = yCabin + cabinHeight;

 **int** xLeftWheel = xEngine + shiftEngineWheels;

 **int** yWheels = yEngine + engineHeight;

 **int** xRightWheel = xEngine + engineWidth - shiftEngineWheels - diameter;

 **int** autosShift = engineWidth + distance;

 **int** roadLayHeight = shiftLineAuto\*2 + cabinHeight + engineHeight +

diameter;

 **public** **void** paint(Graphics painter) {

 painter.drawLine(0, yTopLine, frameLength, yTopLine);

 **for**(**int** lay = 0; lay < laysCount; lay++) {

 **for**(**int** shift = 0; shift < autosCount; shift++) {

 painter.drawRect(xCabin + autosShift \* shift,

 yCabin + roadLayHeight \* lay, cabinWidth, cabinHeight);

 painter.drawRect(xEngine + autosShift \* shift,

 yEngine + roadLayHeight \* lay, engineWidth, engineHeight);

 painter.drawOval(xLeftWheel + autosShift \* shift,

 yWheels + roadLayHeight \* lay, diameter, diameter);

 painter.drawOval(xRightWheel + autosShift \* shift,

 yWheels + roadLayHeight \* lay, diameter, diameter);

 }

 painter.drawLine(0, yTopLine + roadLayHeight \* (lay + 1),

 frameLength, yTopLine + roadLayHeight \* (lay + 1));

 }

 }

}

**Результат:**

****

**Домашнє завдання.** Написати java-код із використанням оператора циклу for для виведення:

- довільної кількості пасажирських вагонів з 9 вікнами (при виведенні коліс та конекторів у вагонах НЕ використовувати оператор for),

- довільної кількості житлових будинків із довільною кількістю поверхів із довільною кількістю вікон на кожному поверсі:



**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 6.**

**Робота з кольорами. Малювання державних прапорів. Робота з текстом (назви держав). Почергове малювання прапорів країн з горизонтальною та вертикальною орієнтацією кольорових смуг.**

**Відео:** 6\_7\_arraysAndFlags.

**Код з відео:**

////// CODE FROM VIDEO “6\_7\_arraysAndFlags” FOR LESSON 6 ///////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** FlagsPainter **extends** Frame {

 FlagsPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(300, 350);

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) {

 FlagsPainter myFlags = **new** FlagsPainter("Flags");

 }

 **int** xStart = 50, yStart = 100;

 **int** flagStockWidth = 10, flagStockHeight = 180;

 **int** flagWidth = 150, flagHeight = 100;

 **int** letterHeight = 35;

 /////////////////////////

 **int** xFlag = xStart + flagStockWidth;

 **int** flagStringHeight = flagHeight / 2;

 **public** **void** paint(Graphics painter) {

 painter.setColor(Color.***ORANGE***);

 painter.fillRect(xStart, yStart, flagStockWidth,

 flagStockHeight);

 painter.setColor(Color.***BLUE***);

 painter.fillRect(xFlag, yStart, flagWidth, flagStringHeight);

 painter.setColor(Color.***YELLOW***);

 painter.fillRect(xFlag, yStart + flagStringHeight, flagWidth,

 flagStringHeight);

 painter.setColor(Color.***BLACK***);

 painter.drawRect(xStart, yStart, flagStockWidth,

 flagStockHeight);

 painter.drawRect(xFlag, yStart, flagWidth, flagHeight);

 painter.setFont(**new** Font("DIALOG", Font.***ITALIC***, letterHeight));

 painter.drawString("UKRAINE", xStart, yStart + flagStockHeight +

 10 + letterHeight);

 }

}

 **Результат:**



**Домашнє завдання.** Написати java-код для виведення державних прапорів:

Польщі, Німеччини, Франції, Швеції, Греції. Рекомендовано використати link:

<https://ru.freepik.com/premium-vector/european-union-countries-flags-europe-travel-states-eu-member-country-flag-set_7394269.htm>

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 7.**

**Масиви (назви держав, кольори одного прапору).**

**Відео:** 6\_7\_arraysAndFlags.

**Код з відео:**

////// CODE FROM VIDEO “6\_7\_arraysAndFlags” FOR LESSON 7 ///////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** FlagsPainter **extends** Frame {

 FlagsPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(400, 400);

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) **throws** Exception {

 FlagsPainter myFlagsPainter = **new** FlagsPainter("Flags");

 }

 **int** xStart = 50, yStart = 100;

 **int** flagStockWidth = 10, flagStockHeight = 180;

 **int** flagWidth = 150, flagHeight = 100;

 //////////////////////////////////////

 Color[] ukraineColors = {Color.***BLUE***, Color.***YELLOW***};

 Color[] polandColors = {Color.***WHITE***, Color.***RED***};

 Color[] germanyColors = {Color.***BLACK***, Color.***RED***, Color.***YELLOW***};

 /////////////////////////

 **int** xFlag = xStart + flagStockWidth;

 **int** flagStringHeight = flagHeight / germanyColors.length;

 **public** **void** paint(Graphics painter) {

 painter.setColor(Color.***LIGHT\_GRAY***);

 painter.fillRect(xStart, yStart, flagStockWidth,

 flagStockHeight);

 **for** (**int** s = 0; s < germanyColors.length; s++) {

 painter.setColor(germanyColors[s]);

 painter.fillRect(xFlag, yStart + flagStringHeight \* s,

 flagWidth, flagStringHeight);

 }

 painter.setColor(Color.***BLACK***);

 painter.drawRect(xStart, yStart, flagStockWidth,

 flagStockHeight);

 painter.drawRect(xFlag, yStart, flagWidth, flagHeight);

 }

}

 **Результат:**



**Домашнє завдання.** Із використанням масивів написати java-код для виведення державних прапорів Болгарії, Литви, Нідерландів, Італії, Румунії, Франції.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 8.**

**Двомірні масиви (кольори прапорів різних країн).**

**Одночасне малювання прапорів різних країн.**

**Відео:** 8\_arraysTwoDimentionals.

**Код з відео:**

//// CODE FROM VIDEO “8\_arraysTwoDimentionals” FOR LESSON 8 ////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** FlagsPainter **extends** Frame {

 FlagsPainter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(frameLength, frameHeight);

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) {

 FlagsPainter painter = **new** FlagsPainter("StateFlagsPainter");

 }

 //// variables for initialization /////////////

 **int** frameHeight = 300, frameLength = 700;

 **int** flagWidth = 180, flagHeight = 120;

 **int** xFlag = 40, yFlag = 40;

 //// variables for calculation /////////////

 **int** distance = flagWidth + 40;

 **int** lineHeight = flagHeight / 3;

 Color[] germany = {Color.***BLACK***, Color.***RED***, Color.***YELLOW***};

 Color[] lithuania = {Color.***YELLOW***, Color.***GREEN***, Color.***RED***};

 Color[] holland = {Color.***RED***, Color.***WHITE***, Color.***BLUE***};

 Color[][] states = {germany, lithuania, holland};

 **public** **void** paint(Graphics painter) {

 **for**(**int** s = 0; s < states.length; s++) {

 **for**(**int** l = 0; l < states[s].length; l++) {

 painter.setColor(states[s][l]);

 painter.fillRect(xFlag + distance \* s,

 yFlag + lineHeight \* l, flagWidth, lineHeight);

 }

 painter.setColor(Color.***BLACK***);

 painter.drawRect(xFlag + distance \* s, yFlag, flagWidth, flagHeight);

 }

 }

}

**Результат:**



**Домашні завдання.**

1. До масиву додати java-код для додаткового виведення державних прапорів Болгарії та Австрії.

2. До масиву додати java-код для додаткового виведення державних прапорів України та Польщі.

3.Окремою програмою із використанням двовимірного масиву написати java-код для виведення державних прапорів Італії, Румунії, Франції.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 9.**

**Робота з кнопками. Опрацювання подій. Зміна при натисканні кнопки:**

**тексту, цілих чисел (інкрементер-декрементер).**

**Відео:** 9\_buttons.

**Код з відео:**

/////////// CODE FROM VIDEO “9\_buttons” FOR LESSON 9 ///////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** Painter **extends** Frame {

 **int** count = 0;

 Painter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(300, 100);

 init();

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) **throws** Exception {

 Painter myButtons = **new** Painter("Buttons");

 }

 **public** **void** init() {

 Panel p = **new** Panel();

 add(p, BorderLayout.***CENTER***);

 Button plus = **new** Button("PLUS");

 Button minus = **new** Button("MINUS");

 TextField tf = **new** TextField("" + count);

 p.add(plus);

 p.add(tf);

 p.add(minus);

 plus.addActionListener(**new** ActionListener() {

 **public** **void** actionPerformed(ActionEvent event) {

 count++;

 tf.setText("" + count);

 }

 });

 minus.addActionListener(**new** ActionListener() {

 **public** **void** actionPerformed(ActionEvent event) {

 count--;

 tf.setText("" + count);

 }

 });

 }

}

**Результат:**

 

**Домашнє завдання.** Написати код простого калькулятора для виконання чотирьох арифметичних дій над двома цілими числами із виведенням результату згідно рисунку:

 

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ 10.**

**Оператор розгалуження if-else. Змінні типу boolean.**

**Інкрементер-декрементер із завданням режиму роботи.**

**Відео:** 10\_if\_else.

**Код з відео:**

////////// CODE FROM VIDEO “10\_if\_else” FOR LESSON 10 //////////

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** Painter **extends** Frame {

 **int** count = 0;

 **boolean** isDecrement = **false**;

 String modeString = "+";

 Painter(String title) {

 **super**(title);

 addWindowListener(**new** WindowAdapter() {

 **public** **void** windowClosing(WindowEvent ev) {

 System.*exit*(0);

 }

 });

 setSize(300, 100);

 init();

 setVisible(**true**);

 }

 **public** **static** **void** main(String[] args) **throws** Exception {

 Painter myButtons = **new** Painter("ButtonsWithMode");

 }

 **public** **void** init() {

 Panel p = **new** Panel();

 add(p, BorderLayout.***CENTER***);

 Button action = **new** Button("ACTION");

 Button modeButton = **new** Button("MODE");

 TextField text = **new** TextField("" + count);

 TextField modeText = **new** TextField(modeString);

 p.add(action);

 p.add(text);

 p.add(modeText);

 p.add(modeButton);

 action.addActionListener(**new** ActionListener() {

 **public** **void** actionPerformed(ActionEvent event) {

 **if**(isDecrement) {

 count--;

 }**else** {

 count++;

 }

 text.setText("" + count);

 }

 });

 modeButton.addActionListener(**new** ActionListener() {

 **public** **void** actionPerformed(ActionEvent event) {

 **if**(isDecrement) {

 modeString = "+";

 }**else** {

 modeString = "-";

 }

 isDecrement = !isDecrement;

 modeText.setText(modeString);

 }

 });

 }

}

**Результат:**

 

**Домашнє завдання.** Написати код простого калькулятора для виконання чотирьох арифметичних дій над двома цілими числами із використанням однієї кнопки завдання арифметичної дії та виведенням результату згідно рисунку:

 

Використати метод equals за прикладом:

String s1 = “\*”:

String s2 = “-”:

 if(s1.equals(s2)){

 // TODO

 }

**ДОДАТОК 2**

**Навчальна програма курсу JavaBasic**

| **№** | **Теми навчальної програми** |
| --- | --- |
| 1 | Робота з Eclipse. Перша програма. Ознайомлення із поняттями клас, пакет, проект. Коментарі. Оголошення змінних. Ідентифікатори. Ключові слова. Типи даних *String, int*. Арифметичні операції. Тип даних *double*. Перетворення (*casting*) типів *int* та *double.* Конкатенація *String*. |
| 2 | Операції порівняння. Тип даних *boolean*. Оператори розгалуження *if* та *if-else.* Логічні функції *NOT, AND, OR.* Перетин прямокутників. |
| 3 | Цикли. Оператор *for*. Область видимості змінної. Вкладені цикли. Оператори *while* та *do-while*. Виведення крапкових трикутників. |
| 4 | Види масивів, способи їх опису. Одновимірні масиви. Поле *length*. Обробка масивів без розгалужень. Оператор *foreach*. Обробка масивів із розгалуженням. Обробка масивів із утворенням масивів-результатів відомої та невідомої довжин. |
| 5 | Ознайомлення із *Java API* , пакетом *java.lang*, класом *String* та методами *length, toLowerCase, toUpperCase, charAt, endWith, startWith* класу *String*. Ознайомлення із поняттям перезавантаження методів та методами *indexOf, replace, split, substring* класу *String*. Ознайомлення із поняттям імпорт, ключовим словом *static* та класом *java.util.Arrays.* Методи *toString, sort, binarySearch* класу *Arrays*. Клас *java.util.ArrayList.* Методи *size, add, get, indexOf* класу *ArrayList*. Обробка масивів цілих чисел із утворенням колекцій *ArrayList*. |

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ ПО ТЕМІ 1**

**1.1 Робота з Eclipse. Перша програма. Ознайомлення із поняттями клас, пакет, проект. Коментарі. Оголошення змінних. Ідентифікатори. Ключові слова. Типи даних *String, int*. Арифметичні операції. Тип даних *double*. Перетворення (*casting*) типів *int* та *double.* Конкатенація *String*.**

**Відео:** 1\_Перша програма.

**Література:** [Шилдт] – с.64…71 (англ. – р.75…79).

**Тести:** [Ganesh] – MockExam1: № 52, 53.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ ПО ТЕМІ 2**

**Операції порівняння. Типи даних *boolean*. Оператори *if* та *if-else.* Логічні функції *NOT, AND, OR.* Використання операторів розгалуження та логічних функцій для розв’язку завдання про перетин прямокутників*.***

**Відео:** 3\_Розгалуження.

**Література:** [Шилдт] – с.85…86, 89, 124…126, 131…135 (англ. – р.98, 99, 102,

 141…143, 150…154),

 [Хорстманн] – c. 93…97 (англ. – р.138…145).

**Тести:** [Heller] -сh.5: № 4,

 [Raposa] - ch.1: № 19,

 [Mughal] - ch.6: № 6.1,

 [Sierra5] - ch.4: № 9.

**Програмне завдання (детальне роз’яснення – у відео “Прямокутники”):**

За допомогою операторів розгалуження та логічних функцій AND, OR визначити тип перетину двох прямокутників R1 та R2 із сторонами, що є паралельними вісям координат. Прямокутники завдано координатами двох діагональних вершин А та В:

int ax1, ay1, bx1, by1 – для прямокутника R1,

int ax2, ay2, bx2, by2 – для прямокутника R2:



Для зменшення громіздкості коду припускаємо, що точка А має бути завжди нижче та лівіше точки В.

Написати Java-код для визначення типу перетину з п’яти можливих варіантів:

– перетину немає,

– перетином є крапка,

– перетином є вертикальна лінія,

– перетином є горизонтальна лінія,

– перетином є прямокутник.

Рекомендовано обрати саме вищенаведену послідовність варіантів, а також кожний фрагмент коду для визначення певного типу перетину оформити у вигляді автономного відокремленого блоку операторів if-else.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ ПО ТЕМІ 3**

**Цикли. Оператор *for*. Область видимості змінної. Вкладені цикли. Оператори *while* та *do-while*. Виведення крапкових геометричних фігур.**

**Відео:** 4\_Цикли.

**Література:** [Шилдт] – с.93 … 95, 140 … 149, 154 … 155 (англ. – р.105 … 108,

 160…171, 178),

 [Хорстманн] – c. 93, 97…103 (англ. – р.139, 140, 145…157).

**Програмне завдання 1:** реалізувати виведення примітивних крапкових трикутників (Шилдт, ст.154-155) нижченаведених конфігурацій, а також прямокутника:

 .

 . .

 . . .

 . . . .

 . . . . .

 . . . . . .

**Програмне завдання 2:** реалізувати виведення примітивних крапкових “горизонтальних” трикутника та трапеції:

**Програмне завдання 3:** із використанням алгоритмів виведення примітивних “вертикальних” трикутників та прямокутника, створених у завданні 2, реалізувати формування складених рівнобедрених “вертикальних”трикутників, паралелограмів та “вертикальних” трапецій наступних конфігурацій:

 **Програмне завдання 4:** із використанням алгоритму виведення примітивного “горизонтального” трикутника, створеного у завданні 3, реалізувати виведення крапкових трикутників, розміщених вертикально (“ялинку”):

Передбачити можливість побудови “ялинки” з трикутників різного розміру та з різними символами формування (“крапками”), зміну кількості трикутників для формування “ялинки”, а також можливість зсуву “ялинки” вправо на задану кількість кроків.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ ПО ТЕМІ 4**

**Види масивів, способи їх опису. Одновимірні масиви. Поле *length*. Обробка масивів без розгалужень. Обробка масивів із розгалуженням. Обробка масивів із утворенням масивів-результатів відомої та невідомої довжин.**

**Відео:** 5\_Масиви.

**Література:** [Шилдт] – с.100…102, 107 (англ. – р.113…115, 121, 122),

 [Хорстманн] – c. 111…113 (англ. – р.168…171).

**Тести:** [Heller] -Assesment Test: 11,

 [Jaworsky] - p.71: 20.

**Програмні завдання опрацювання масивів цілих чисел (припускаємо, що елементи в масивах не повторюються):**

- порахувати кількість та суму парних елементів масиву,

- визначити максимальний з від’ємних та мінімальний з додатніх елементів масиву,

- збільшити на одиницю додатні елементи масиву та зменшити на одиницю від’ємні елементи масиву із утворенням нового масиву,

- в масиві поміняти місцями максимум та мінімум із утворенням нового масиву,

- в масиві усунути максимум та мінімум із утворенням нового масиву,

- визначити середнє арифметичне елементів масиву та кількість елементів цього масиву, що є більшими за середнє арифметичне.

**САМОСТІЙНА ПІДГОТОВКА ДО ЗАНЯТТЯ ПО ТЕМІ 5**

**Ознайомлення із *Java API* , пакетом *java.lang*, класом *String* та методами *length, toLowerCase, toUpperCase, charAt, endsWith, startsWith* класу *String*. Ознайомлення із поняттям перезавантаження методів та методами *indexOf, replace, split, substring* класу *String*. Ознайомлення із поняттям імпорт, ключовим словом *static* та класом *java.util.Arrays.* Методи *toString, sort, binarySearch* класу *Arrays*. Клас *java.util.ArrayList.* Методи *size, add, get, indexOf, contains* класу *ArrayList*. Обробка масивів цілих чисел із утворенням колекцій *ArrayList*.**

**Відео:** 8\_АРІ класів String, Arrays, ArrayList.

**Література:** [Шилдт] – с.536, 539, 540, 542…549 (англ. – р.652, 656, 659…667),

 [Хорстманн] – c. 72, 73, 77…79 (англ. – р.104…106, 113…116).

**Тести:** [Raposa] – ch.7: 4, 14,

 [Jaworsky] – p.157: 15,

 [Reese] – ch.2: 5,

 [Boyarsky] – ch.3: 7, 8,

 [Ganesh] – ch.7: 1,

 MockExam2: 18,

 [MalaGupta] – MockExam: 44,

 p.186 -Tale 4.1,

 [Sierra5] – ch.7: 9,

 [Sierra7] – ch.6: 1.

**Програмні завдання.**

**1. Опрацювання масивів цілих чисел із утворенням колекцій:**

1.1. Масив розділити на дві колекції з від’ємних та додатніх чисел,

1.2. Визначити середні арифметичні двох масивів, утворити колекцію з елементів обидвох масивів, що знаходяться в межах між значеннями середніх арифметичних,

1.3. Утворити колекцію з масиву, в якому видалити дублікати максимума та мінімума (уточнення: видалити саме дублікати, тобто перші значення максимума та мінімума необхідно залишити, інший варіант – залишити останні значення максимума та мінімума).

 **2. Опрацювання Strings. Для всіх завдань ігнорувати регістр букв - строкові чи прописні. В нижченаведених завданнях утворювати новий текстовий рядок, не змінюючи початкового:**

2.1. Зробити великими перші букви у кожному слові тексту,

2.2. В тексті слова заданої довжини заміниті вказаним підрядком довільної довжини,

2.3. Після кожного слова тексту, що закінчується заданим під­рядком, вставити вказане слово,

2.4. В тексті до кожного слова з довжиною, більшою за k символів, вставити заданий підрядок після k-го символу,

2.5. Утворити колекцію зі слів тексту, в яких перша і остання букви співпадають,

2.6. Утворити дві колекції зі слів тексту максимальної та слів мінімальної довжини,

2.7. В кожному реченні тексту поміняти місцями перше слово з останнім (припускаємо, що всі речення закінчуються тільки крапкою).