**Soal Pertama**Apa perbedaan paling signifikan yang menyebabkan *Checked Exception* dan *Unchecked Exception* ? serta sebutkan beberapa Class Exceptionnya !

**Jawaban**

Kalau Checked Exception itu tidak dapat dibiarkan begitu saja, kita harus menangani kesalahan ini,

Kalau Checked Exception itu tidak harus ditangani

Class Checked Exception = Class Non Found Exception, Clone Not Supported Exception, Illegal Eccess Exception , Instantiation Exception, Inteerruption Exception, No Such Field Exception, No Such Method Exception, IO Exception, AWT Exception

Class Unchecked Exception = Arithem Exception, Array Index Out of Bounds Exception, Array Store Exception, Class Cast Exception, Illegal Argument Exception, Illegal Momitor Store Exception, Illegal State Exception, Illegal Thread State Exception, Index Out Of Bounds Exception, Negative Array Size Exception, Null Pointer Exception, Number Format Exception

**Soal Kedua**Cici ingin membuat sebuah program kalkulator sederhana yang sedikit unik, yang dimana jika user memasukan String “2+2” maka program dapat menghitung hasilnya adalah 4

|  |
| --- |
| 1. **import** java.util.Scanner; 3. **public** **class** kalkulator { 4. **public** **static** **void** main(String[] args) { 6. Scanner input = **new** Scanner(System.in); 8. System.out.println("Input Operasi"); 9. String angkaStr = input.nextLine(); 11. String[] inputUser = angkaStr.split(" "); 13. **int** angka1 = Integer.parseInt(inputUser[0]); 14. String operator = inputUser[1]; 15. **int** angka2 = Integer.parseInt(inputUser[2]); 17. **switch** (operator) { 18. **case** "+": 19. **int** hasil = angka1 + angka2; 20. System.out.println(("Hasil = " + hasil)); 21. **break**; 22. } 24. } 25. } |

Setelah programnya selesai, cici pun iseng mencoba memasukan input seperti ini “2 + dua”. Dan tiba-tiba programnya langsung berhenti karena terjadi Exception.

Bantulah cici untuk memperbaiki programnya supaya program miliknya bisa menampilkan pesan kesalahan jika terjadi hal yang serupa yang telah ia alami, cobalah menambahkan Exception Handling yang tepat pada program diatas!

**Jawaban**

|  |
| --- |
| 1. **import** java.util.Scanner; 3. **public** **class** kalkulator { 4. **public** **static** **void** main(String[] args) { 5. **try** { 6. Scanner input = **new** Scanner(System.in); 8. System.out.println("Input Operasi"); 9. String angkaStr = input.nextLine(); 11. String[] inputUser = angkaStr.split(" "); 13. **int** angka1 = Integer.parseInt(inputUser[0]); 14. String operator = inputUser[1]; 15. **int** angka2 = Integer.parseInt(inputUser[2]); 17. **switch** (operator) { 18. **case** "+": 19. **int** hasil = angka1 + angka2; 20. System.out.println(("Hasil = " + hasil)); 21. **break**; 22. } 23. } **catch** (Exception e) { 24. System.out.println("Terdapat kesalahan pada : " + e); 25. } 26. } 27. } |

**Output**

|  |
| --- |
|  |

**Soal Ketiga**Pada Frame yang menggunakan Java Swing, pada Constructornya kita pasti memnaggil sebuah method *setLayout(null)*. Apakah yang akan terjadi ?

**Jawaban**

Ketika tidak memanggil method setLayout(null) maka tidak akan muncul warna pada permukaan GUI kita, dan layoutnya tidak muncul dan tata letaknya tidak tertata, dan semuanya di rata kiri.

**Soal Keempat**

Buatlah sebuah kode Program Java Swing yang memiliki tampilan seperti berikut :

|  |
| --- |
|  |
| 1  +1  x2  -3  -1  +3 |

Terapkan konsep ActionListener untuk setiap Button yang ada !

**Jawaban**

|  |
| --- |
| 2. **import** javax.swing.\*; 3. **import** java.awt.\*; 4. **public** **class** CobaGUI **extends** JFrame { 5. **private** JFrame cobaGUI; 6. **private** JLabel middle; 7. **private** JLabel nomerLabel; 8. **private** JButton b1, b2, b3, b4, b5; 9. **private** **int** nomer; 10. **public** CobaGUI(){ *//FRAME TAMPILAN GUI* 11. setTitle("Nomer 4 GUI"); 12. setSize(800,600); 13. getContentPane().setBackground(Color.gray); 14. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); 15. setLayout(**null**); 17. component(); 18. event(); 19. } 21. **private** **void** component(){ *//ISI GUI* 22. String penjumlahan = String.valueOf(nomer); 23. middle = **new** JLabel(penjumlahan);*// Text TENGAH* 24. middle.setBounds(342,60 ,100,50); 25. *//bounds = ukuran maks* 26. *//x(Jarak Kiri), y(Jarak atas), width(Lebar), height(tinggi)* 27. middle.setFont(**new** Font("Times New Roman",Font.BOLD,40)); 28. add(middle); 30. *// Atur Lokasi Button* 31. b1 = **new** JButton("+1"); 32. b1.setBounds(100,130,200,100); 33. b1.setBackground(Color.WHITE); 34. add(b1); 36. b2 = **new** JButton("-1"); 37. b2.setBounds(100,290,200,100); 38. b2.setBackground(Color.WHITE); 39. add(b2); 41. b3 = **new** JButton("x2"); 42. b3.setBounds(242,430,200,100); 43. b3.setBackground(Color.WHITE); 44. add(b3); 46. b4 = **new** JButton("+3"); 47. b4.setBounds(400,130,200,100); 48. b4.setBackground(Color.WHITE); 49. add(b4); 51. b5 = **new** JButton("-3"); 52. b5.setBounds(400,290,200,100); 53. b5.setBackground(Color.WHITE); 54. add(b5); 56. } 58. **private** **void** event(){ 59. b1.addActionListener((event) -> { 60. nomer +=1; 61. String penjumlahan = String.valueOf(nomer); 62. middle.setText(penjumlahan); 63. }); 64. b2.addActionListener((event) -> { 65. nomer -=1; 66. String penjumlahan = String.valueOf(nomer); 67. middle.setText(penjumlahan); 68. }); 69. b3.addActionListener((event) -> { 70. nomer \*=2; 71. String penjumlahan = String.valueOf(nomer); 72. middle.setText(penjumlahan); 73. }); 74. b4.addActionListener((event) -> { 75. nomer +=3; 76. String penjumlahan = String.valueOf(nomer); 77. middle.setText(penjumlahan); 78. }); 79. b5.addActionListener((event) -> { 80. nomer -=3; 81. String penjumlahan = String.valueOf(nomer); 82. middle.setText(penjumlahan); 83. }); 85. } 87. **public** **static** **void** main(String[] args) { 88. CobaGUI cobaGUI = **new** CobaGUI(); 89. cobaGUI.setVisible(**true**); 90. } 91. } |

**Output**

|  |
| --- |
|  |

**Soal kelima**

***PROGRES STUDI KASUS PROJECT AKHIR:***

Pada program studi kasus kalian tambahkan beberapa Exception Handling (Checked Exception, Unchecked Exception, atau Error) pada bagian kode program kalian yang memungkinkan terjadi Exception, dan ubahlah tampilan CLI project kalian menjadi sebuah tampilan Graphical User Interface (GUI), gunakan Java Swing untuk membuat Tampilan GIU-nya !

**Jawaban**

**Source Code**

AkunController

|  |
| --- |
| 1. **package** controller; 3. **import** entity.DaftarAkunEntity; 5. **public** **class** AkunController{ 6. **int** indexlogin = 0; 7. **public** AkunController(){ 9. } 10. **public** DaftarAkunEntity getData(){ 11. **return** AllObjectModel.daftarAkunModel.showDataAKun(indexlogin); 12. } 13. **public** **void** daftarKursus(String idakun, String nama, String alamat, String notlp, String pw, String kursus){ 14. AllObjectModel.daftarAkunModel.insertDataDaftarAkun(**new** DaftarAkunEntity(idakun,nama,alamat,notlp,pw,kursus)); 15. } 17. **public** **void** login(String idakun, String pw){ 18. indexlogin = AllObjectModel.daftarAkunModel.cekData(idakun,pw); 19. } 20. **public** **void** viewAkun(){ 21. AllObjectModel.daftarAkunModel.view(); 22. } 24. **public** **void** hapusbuku(**int** index){ 25. AllObjectModel.daftarAkunModel.delete(index-1); 26. } 27. **public** **void** editakun(**int** index, **int** pilih, String data){ 28. AllObjectModel.daftarAkunModel.update(index-1,pilih,data); 29. } 31. } |

AllObjectModel

|  |
| --- |
| 1. **package** controller; 3. **import** model.DaftarAkunModel; 5. **public** **class** AllObjectModel { 6. **public** **static** DaftarAkunModel daftarAkunModel = **new** DaftarAkunModel(); 7. } |

DaftarAkunEntity

|  |
| --- |
| 1. **package** entity; 3. **public** **class** DaftarAkunEntity { 4. **private** String idakun; 5. **private** String nama , alamat , notlp , pw , kursus; 7. **public** String getKursus() { 8. **return** kursus; 9. } 11. **public** **void** setKursus(String kursus) { 12. **this**.kursus = kursus; 13. } 15. **public** DaftarAkunEntity(String idakun, String nama, String alamat, String notlp, String pw, String kursus) { 16. **this**.idakun = idakun; 17. **this**.nama = nama; 18. **this**.alamat = alamat; 19. **this**.notlp = notlp; 20. **this**.pw = pw; 21. **this**.kursus = kursus; 22. } 24. **public** String getIdakun() { 25. **return** idakun; 26. } 28. **public** **void** setIdakun(String idakun) { 29. **this**.idakun = idakun; 30. } 32. **public** String getNama() { 33. **return** nama; 34. } 36. **public** **void** setNama(String nama) { 37. **this**.nama = nama; 38. } 40. **public** String getAlamat() { 41. **return** alamat; 42. } 44. **public** **void** setAlamat(String alamat) { 45. **this**.alamat = alamat; 46. } 48. **public** String getNotlp() { 49. **return** notlp; 50. } 52. **public** **void** setNotlp(String notlp) { 53. **this**.notlp = notlp; 54. } 56. **public** String getPw() { 57. **return** pw; 58. } 60. **public** **void** setPw(String pw) { 61. **this**.pw = pw; 63. } 64. } |

DaftarAkunModel

|  |
| --- |
| 1. **package** model; 3. **import** entity.DaftarAkunEntity; 5. **import** java.util.ArrayList; 7. **public** **class** DaftarAkunModel { 8. **private** ArrayList<DaftarAkunEntity> daftarAkunEntityArrayList; 9. **public** DaftarAkunModel(){ 10. daftarAkunEntityArrayList= **new** ArrayList<DaftarAkunEntity>(); 11. } 12. **public** **void** insertDataDaftarAkun(DaftarAkunEntity daftarakun){ 13. daftarAkunEntityArrayList.add(daftarakun); 14. } 15. **public** ArrayList<DaftarAkunEntity> getDaftarAkunEntityArrayList(){ 16. **return** daftarAkunEntityArrayList; 17. } 18. **public** ArrayList<DaftarAkunEntity> alldatakursus(){ 19. **return** daftarAkunEntityArrayList; 20. } 22. **public** **void** view(){ 23. **for**(DaftarAkunEntity daftarAkun : daftarAkunEntityArrayList){ 24. System.out.println("======================"); 25. System.out.println("ID Akun :"+daftarAkun.getIdakun()); 26. System.out.println("Nama depan : "+daftarAkun.getNama()); 27. System.out.println("Alamat : "+daftarAkun.getAlamat()); 28. System.out.println("Nomer Telephone : "+daftarAkun.getNotlp()); 29. System.out.println("Password: "+daftarAkun.getPw()); 30. System.out.println("Kursus: "+daftarAkun.getKursus()); 31. } 32. } 34. **public** **int** cekData(String idakun, String pw){ 35. **int** loop = 0; 36. **for**(DaftarAkunEntity daftarAkunEntity : daftarAkunEntityArrayList){ 37. **if** (daftarAkunEntity.getIdakun().equals(idakun) && daftarAkunEntity.getPw().equals(pw)){ 38. **break**; 39. } **else** { 40. loop++; 41. } 42. } 43. **return** loop; 44. } 46. **public** **int** search (**int** index) 47. { 48. **if** (index < **this**.daftarAkunEntityArrayList.size()) 49. **return** index; 50. **else** 51. **return** -1; 52. } 54. **public** **void** delete(**int** index) 55. { 56. **if** (search(index)==-1) 57. { 58. System.out.println("data tidak ditemukan"); 59. } 60. **if** (index == -1) 61. { 62. System.out.println("data tidak ditemukan"); 63. } 64. **else** 65. { 66. daftarAkunEntityArrayList.remove(index); 67. System.out.println("berhasil dihapus"); 68. } 69. } 71. **public** **void** update( **int** index, **int** pilih, String data){ 72. **if** (search(index)==-1){ 73. System.out.println("Data tidak ditemukan"); 74. }**else**{ 75. **switch** (pilih){ 76. **case** 1: 77. daftarAkunEntityArrayList.get(index).setIdakun(data); 78. **break**; 79. **case** 2: 80. daftarAkunEntityArrayList.get(index).setNama(data); 81. **break**; 82. **case** 3: 83. daftarAkunEntityArrayList.get(index).setAlamat(data); 84. **break**; 85. **case** 4: 86. daftarAkunEntityArrayList.get(index).setNotlp(data); 87. **break**; 88. **case** 5: 89. daftarAkunEntityArrayList.get(index).setPw(data); 90. **break**; 91. **case** 6: 92. daftarAkunEntityArrayList.get(index).setKursus(data); 93. **break**; 94. **default**: 95. } 96. } 97. } 98. **public** DaftarAkunEntity showDataAKun (**int** index){ 99. **return** daftarAkunEntityArrayList.get(index); 100. } 101. } |

Akun

|  |
| --- |
| 1. **package** view; 3. **import** controller.AllObjectModel; 4. **import** model.DaftarAkunModel; 6. **import** java.util.Scanner; 8. **public** **class** Akun { 10. **private** **static** Scanner input = **new** Scanner(System.in); 12. **public** **static** **void** Daftar(){ 13. String idakun; 14. String nama , alamat , notlp , pw; 15. **int** indexkursus = 0; 16. System.out.print("ID : "); 17. idakun = input.nextLine(); 18. System.out.print("Nama : "); 19. nama = input.nextLine(); 20. System.out.print("Alamat : "); 21. alamat = input.nextLine(); 22. System.out.print("Nomer telephone : "); 23. notlp = input.nextLine(); 24. System.out.print("password : "); 25. pw = input.nextLine(); 26. System.out.println("Menu Kursus : "); 27. System.out.println("- Matematika "); 28. System.out.println("- Biologi "); 29. System.out.println("- Fisika "); 30. System.out.print("Pilih Kursus : "); 31. String kursus = input.next(); 32. AllObjectController.akunController.daftarKursus(idakun,nama,alamat,notlp,pw,kursus); 33. *// AllObjectController.kursusController.dataKursus(kursus);* 35. } 37. **public** **static** **void** login(){ 38. System.out.print("id : "); 39. String idakun = input.next(); 40. System.out.print("Passowrd : "); 41. String pw = input.next(); 42. **try**{ 43. AllObjectController.akunController.login(idakun, pw); 44. menudalam(); 45. } **catch** (Exception e){ 46. System.out.println("Daftar Kursus tidak ada"); 47. } 49. } 50. **public** **static** **void** menu(){ 51. **int** pilih; 52. **do** { 53. System.out.println("Menu Pilihan"); 54. System.out.println("1. Daftar Akun"); 55. System.out.println("2. Login "); 56. System.out.println("3. Exit"); 57. System.out.print("pilih : "); 58. pilih = input.nextInt(); 59. input.nextLine(); 60. **switch** (pilih){ 61. **case** 1: 62. Daftar(); 63. **break**; 64. **case** 2: 65. login(); 66. **break**; 67. } 68. } **while** (pilih != 3); 69. } 71. **public** **static** **void** menudalam(){ 72. **int** pilih; 73. **do** { 74. System.out.println("Menu Pilihan"); 75. System.out.println("1. Edit "); 76. System.out.println("2. Hapus"); 77. System.out.println("3. lihat "); 78. System.out.println("4. Exit"); 79. System.out.print("pilih : "); 80. pilih = input.nextInt(); 81. input.nextLine(); 82. **switch** (pilih){ 83. **case** 1: 84. editbuku(); 85. **break**; 86. **case** 2: 87. deletebuku(); 88. **break**; 89. **case** 3: 90. AllObjectController.akunController.viewAkun(); 91. **break**; 92. } 93. } **while** (pilih != 4); 94. } 96. **public** **static** **void** deletebuku(){ 97. AllObjectController.akunController.viewAkun(); 98. System.out.println("Masukkan nomor akun yang ingin di hapus : "); 99. **int** nomor = input.nextInt(); 100. input.nextLine(); 101. AllObjectController.akunController.hapusbuku(nomor); 102. } 103. **public** **static** **void** editbuku(){ 104. **int** pilih; 106. System.out.print("nomor : "); 107. **int** nomor = input.nextInt(); 108. input.nextLine(); 109. System.out.println("Menu Pilihan"); 110. System.out.println("1. ID Akun "); 111. System.out.println("2. Nama"); 112. System.out.println("3. alamat"); 113. System.out.println("4. nomer telephone "); 114. System.out.println("5. password"); 115. System.out.println("6. Kursus"); 116. System.out.print("pilih : "); 117. pilih = input.nextInt(); 118. input.nextLine(); 119. System.out.println("Masukan Data Baru"); 120. String data = input.nextLine(); 121. **switch** (pilih){ 122. **case** 1: 123. AllObjectController.akunController.editakun(nomor,1,data); 124. **break**; 125. **case** 2: 126. AllObjectController.akunController.editakun(nomor,2,data); 127. **break**; 128. **case** 3: 129. AllObjectController.akunController.editakun(nomor,3,data); 130. **break**; 131. **case** 4: 132. AllObjectController.akunController.editakun(nomor,4,data); 133. **break**; 134. **case** 5: 135. AllObjectController.akunController.editakun(nomor,5,data); 136. **break**; 137. **case** 6: 138. AllObjectController.akunController.editakun(nomor,6,data); 139. **default**:  142. }  145. } 146. } |

AllObjectController

|  |
| --- |
| 1. **package** view; 3. **import** controller.AkunController;  6. **public** **class** AllObjectController { 7. **public** **static** AkunController akunController = **new** AkunController(); 8. } |

**Output Program**

|  |
| --- |
|  |