TEST 1

1. Zašto je uvedena Java swing podrška?

- Zato što AWT nije pogodan paket za složenije zahteve i ima bag-ove pri primeni na pojedinim platformama. **Swing** je paket koji omogućava izradu robusnijih, pouzdanijih i složenijih grafičkih komponenti za komunikaciju korisnika sa nekom aplikacijom.

2. Objasnite strukturu Swing paketa?

- Hijerarhijska struktura Swing paketa obuhvata sledeće: „Klasa **Container** je podklasa klase **Component** koja ima tri podklase: **Panel**, **Window** i **JComponent**. Podklase klase Window su **JFrame** i **JDialog** koje prikazuju okvir unutar koga se prikazuju ostale GUI komponente. Klasa **Panel**, tj. njena podklasa **JApplet** predstavlja ploču na kojoj se prikazuju komponente klase JComponent. Klasa JComponent ima više podklasa za prikazivanje različitih komponenti.

3. Koje kontejnerske klase poznajete?

- Klase: Window, Frame, Dialog, JFrame, JDialog, ...

4. Navedite i objasnite pomoćne GUI klase?

- Pomoćne GUI klase su:

1. **Graphics** – za crtanje stringova, linija I jednostavnih objekata,
2. **Color** – za podešavanje GUI komponenata,
3. **Font** – za tekstualn fontove I crteže u GUI komponentama,
4. **FontMetrics** – daje inf. o svojstvima fontova,
5. **LayoutManager** – određuje raspored komponenata u kontejneru.

5. Objasnite ulogu klase Jframe?

- Da bi se kreirao korisnički interfejs, neophodno je da se kreira okvir ili aplet koji bi sadržao sve komponente interfejsa. Za kreiranje okvira koristi se klasa **JFrame**. Okvir je praktično objekat klase JFrame.

6. Koje menadžere rasporeda poznajete?

- Postoje 3 menadžera rasporeda:
  1. **FlowLayout**,
  2. **GridLayout**,
  3. **BorderLayout**.

7. Opišite poznate menadžere rasporeda?

- **FlowLayout** je najjednostavniji menadžer sadržaja. On ređa komponente u kontejneru s leva udesno, po redosledu njihovog ddavanja. Kada se popuni jedan red, stvara se novi red i tako redom.

- **GridLayout** postavlja komponente po ćelijama jednake veličine na rešetki. On deli kontejner u rešetku a komponente dopdavanjem popunjavaju ćelije rešetke rd po red.

- **BorderLayout** deli kontejner na 5 delova: istočni (East), južni (South), zapadni (West), severni (North) i centralni (Center). Komponente se dodaju u kontejner primenom metoda add(). Komponente se rasporedjuju u skladu sa njihovim poželjnim veličinama i naznačenom delu kontejnera.

8. Objasnite ulogu JPanel klase u GUI programima?

- Klasa **JPanel** se u Swing paketu koristi za rad sa panelima. **Paneli** se koriste kao podkontejneri radi grupisanja GUI komponenti da bi se ostvario željeni raspored.

9. Objasnite ulogu Font klase u GUI programima?

- Klasa **Font** se koristi za kreiranje i podešavanje svojstava fontova GUI komponenata. Fontovi su objekti kreirani pomoću klase Font.

10. Objasnite ulogu Color klase u GUI programima?

- Klasa **Color** se koristi za podešavanje boja GUI komponenata. Boje su objekti kreirani pomoću klase Color. Svaka boja je kombinacija: RGB – Red, Green, Blue.

11. Koji zadatak obavljaju klase Component, Container i JComponent? ***

- Klase (superklase) **Component**, **Container** i **JComponent** definišu zajednička svojstva GUI komponenata.

12. Šta je ikona?

- **Ikona** je slika fiksne veličine koja se može prikazati kod mnogih GUI komponenata. Ikone slika su objekti kreirani upotrebom klase **ImageIcon**.

13. Koje formate slika podržava Java Swing?

- Podržava: **GIF**, **JPEG** i **PNG** formate.

14. Dajte primer korišćenja klase JButton?

- ImageIcon usIcon = new ImageIcon(“image/usIcon.gif“);

15. Dajte primer koriscenja klase JCheckBox?

- JCheckBox jchk = new JCheckBox(“Student”, true);

16. Dajte primer upotrebe klase JradioButton?

- JRadioButton jrb = new JRadioButton(“Student”, true);

17. Ako biste kreirali pitanje sa više opcija i jednim tačnim odgovorom koju kontrolu biste koristili: JCheckBox ili JRadioButton?

- JRadioButton.

18. Gde biste najčešće koristili Labele? ***

- Klasa **JLabel** se koristi za kreiranje natpisa u korisničkom interfejsu.

19. Dajte primer upotrebe klase JLabel?

- ImageIcon icon = new ImageIcon(“image/grapes.gif“);
  JLabel jlbl = JLabel(“Grapes”, icon, JLabel.CENTER);

20. Dajte primer uoptrebe klsae JtextField_

- JTextField jtfMessage = new JTextField(“T-Storm”);

21. Koja je suštinska razlika između labela i polja za unos teksta? ***

- Razlika je u tome što se labele koriste za prikaz nekog gotovog ili kreiranog natpisa a polje za unos teksta se koristi za prijem podataka unetih od strane korisnika.

TEST 2

1. Koji zadatak ima klasa Graphics u GUI programima?

   - **Klasa Graphics** je apstraktna klasa obezbeđuje metode za ispisivanje tekstova, linija, pravougaonika, elipsi, lukova, poligona i polilinija. Ovi metodi omogućavaju da nactramo grafičke elemente koje koriste GUI komponente.

2. Koja metoda je zadužena za crtanje linija?

   - Metodu **drawLine()** koristimo za crtanje linija.
     Koristi se u formi: drawLine(int x1, int y1, int x2, int y2).

3. Kako biste nacrtali pravougaonik 20 x 30? ***

   - Java obezbeđuje 6 metoda za crtanje pravougaonika i to: drawRect(), fillRect(), drawRoundRect(), fillRoundRect(), draw3DRect(), fill3DRect(). Da bih nacrtao pravougaonik 20 x 30 koristim: **drawRect(int x, int y, int w, int h);**

   drawRect(10, 10, 30, 20);

4. Koja metoda je zadužena za crtanje lukova? ***

   - Metoda: **drawArc()**.
     Koristi se u formi: drawArc(int x, int y, int w, int h, int startAngle, int arcAngle);

5. U kojim smerovima je moguće crtati lukove?

   - Moguće ih je crtati u smeru skazaljke na satu (negativni uglovi) i u suprotnom smeru (pozitivni uglovi).

6. Koji zadatak obavlja klasa Polygon? ***

   - Klasa Polygon je klasa koja se koristi za crtanje poligona. Ona sadrži metodu **drawPolygon()** koja se upravo koristi za crtanje poligona.

7. Koje metode se primenjuju za crtanje i bojenje poligona?

   - Za crtanje: **drawPolygon()**, a za bojenje: **fillPolygon()**.

8. Koji zadatak obavlja klasa FontMetrics?

- **FontMetrics** je apstraktna klasa koja se koristi za određivanje atributa i centriranje teksta. Ona meri atribute teksta kao što su Leading, Ascent, Decent i Height, koji predstavljaju veličine (tj. metriku) za određeni font.

9. Objasnite kako su u primeru prikazane slike?

- Slike možemo prikazati u grafičkom kontekstu. Praktično, možemo da kreiramo ikonu sa slikom I da je ubacimo u naš korisnički interfejs po potrebi. Slika se kreira metodom **getImage()**.
Metod **drawImage()**, omogućava da se objekat Graphics prikaže kao slika na nekoj GUI komponentu, jer on prikazuje sliku učitanu preko odgovarajuće datoteke, koja popunjava ceo panel.
Pored ovoga, i klasa **ImageViewer** prikazuje određenu sliku na nekom panelu a metoda **paintComponent()** prikazuje sliku na panelu.

TEST 3

TEST 4

TEST 5

TEST 6

1. Šta je jedinično testiranje?

- **Jedinično testiranje** je testiranje pojedinačnih objekata.

2. Šta je JUnit?

- **JUnit** je framework (tj. skup alata) koji omogućava automatizovano testiranje metoda i klasa u fazi razvoja softvera.

3. Navedite prednosti testiranja softvera?

- **Prednosti** su:
  - Veći kvalitet koda,
  - Lakše čitanje koda,
  - Proces testiranja se lako uklapa u proces izrade softvera,
  - Omogućava pravljenje testova prihvatanja (Acceptance testing),
  - Omogućava refaktoring koda,
  - Smanjuje količinu grešaka (Bug-ova) u kodu,
  - Povećava pouzdanost celog sistema.

4. Navedite nedostatke testiranja softvera? ***

- **Nedostaci** su:
  - Ne može da se primeni na GUI klase,
  - Testiraju se samo klase a ne i okruženje u kome se radi,
  - Kod mora da bude rađen po strogim OOP principima,
  - Sporiji je razvojni ciklus softvera,
  - Piše se i do 3 puta više koda u okviru testa, nego što ima u samom kodu koji se izvršava.

5. O čemu je neophodno voditi računa prilikom izrade JUnit testova? ***

- Prilikom jediničnog testiranja, najviše se obraća pažnja na **metode** koje nešto rade, tj. implementiraju neki algoritam. Ovo su metode koje određuju funkcionalnost naše aplikacije, pa je to upravo i mesto koje moramo najviše da testiramo. Takođe, pored metoda, testiraju se i **konstruktori**, ako se u okviru njih generiše nekakav funkcionalni kod.

6. Šta JUnit test treba da proveri?

- Jediničnim testiranjem se proveravaju:

- Da li aplikacija daje tačan rezultat za unete realne vrednosti,
- Potencijalno problematične vrednosti (npr. deljenje sa nulom),
- Granične vrednosti,
- Vrednosti koje su sigurno pogrešne,
- Format dobijenog rezultata,
- Slučajevi kada fali podatak.

7. Koje klase se koriste u JUnit testovima? ***

- **Klase** koje se koriste su:

- TestCase,
- TestSuite,
- TestRunner,
- TestResults.

8. Šta su assert metode i kako se koriste?

- **Assert metode** služe da se proveri ispravnosti neke vrednosti prostih tipova i za proveru objekata. Koriste se tako što se pomoću njih vrši provera tvrdnji u programima.

9. Kako se koriste pravila testiranja?

- **Pravila @Rule**, dozvoljavaju testerima fleksibilnu primenu definicije ponašanja svakog test metoda u test klasi. Testeri mogu da koriste ranije definisana pravila ili da ih proširuju ili da definišu nova pravila.