

## ПРОЕКТИРАНЕ НА ГПИ

1. За извеждането на съобщения е удобно да се използват **етикети**, а за въвеждане на името на потребителя – **текстово поле**.
2. За задаване на функционалност се използват **бутони** за поздрав; за изчистване на съдържанието; за изход.

### СТЪПКА ПО СТЪПКА – създаване на форма

1. Създайте ново приложение **MeetJava**.
2. В папката **src** добавете пакет **edu.school**.
3. В пакета създайте нов **JFrame** и го именувайте **MeetJavaFrame**.
4. Загйте заглавие (*title*) **Запознай се с Java!**

### СТЪПКА ПО СТЪПКА – контроли

5. Разположете контролите върху формата.
6. Преименувайте всички контроли (**Properties** → **Variable**).
7. Променете текстовете на етикетите и бутоните.
8. Създайте естетичен дизайн, като зададете шрифт, размер и цвят на контролите.

### АНАЛИЗ — бутон **Запознай се!**

1. За запазване на името на потребителя е необходима променлива:  
`String name;`
2. Променливата получава стойност от текстово поле.
3. За всяко от двете съобщения въведено то име се **залепва** към останалата част от текста.
4. Съобщенията се извеждат в етикети.

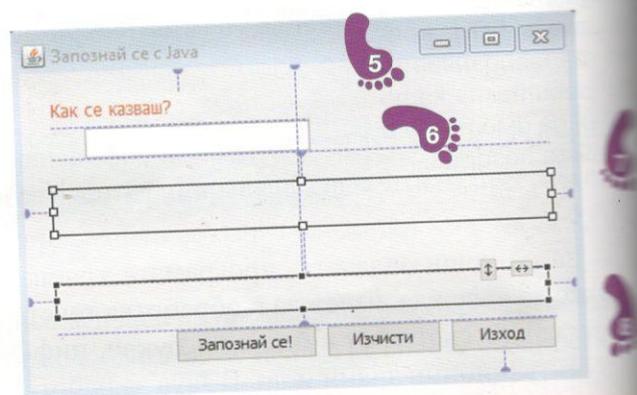
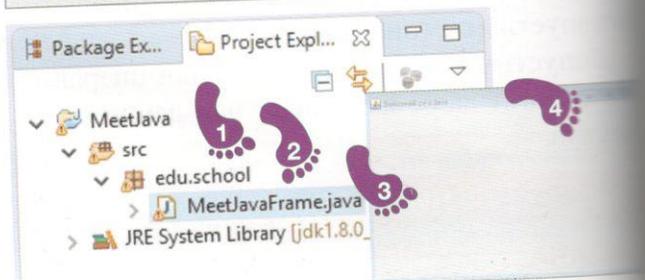
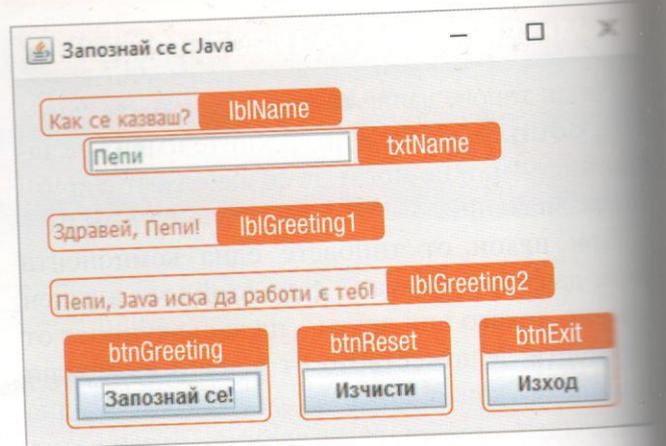
## ПРОМЕНЛИВИ

**Конкатенацията** е операция, при която се слепват два или повече низа. Представя се чрез знака **+**, поставен между аргументите.

### ПРИМЕР – Конкатенация на низове

`"Здравей, " + name + "!"`

Ако променливата **name** има стойност „Пену“, то резултатът ще бъде „Здравей, Пену!“.



## ВЪВЕЖДАНЕ И ИЗВЕЖДАНЕ НА НИЗ

За въвеждане на низ от текстово поле се използва метода **getText()** на контролата.

За извеждане на текст в текстово поле или етикет се използва метода **setText(str)** на контролата, където **str** е символният низ, който се извежда.

### ПРИМЕР – Инициализация чрез въвеждане

```
String name = txtName.getText();
```

Променливата **name** се инициализира със стойността, въведена в текстовото поле.

### ПРИМЕР – Извеждане в контрола

```
lblGreeting1.setText("Здравей!");  
txtName.setText("Г-н/ца" + name);
```

В етикета **lblGreeting1** се извежда текстът „Здравей!“, в **txtName** – резултатът от конкатенацията.

## ПРИСВОЯВАНЕ НА СТОЙНОСТ

Промениливите могат да получават стойност директно при инициализация, но и чрез **присвояване на стойност**. Осъществява се чрез знака =.

Операцията е **дясноасоциативна**: първо се присвоява стойността на израза в дясната част на знака =, след което тази стойност се записва в променливата.

Може да се присвоява стойност само на променливи, които вече са декларирани!

Ако променливата е имала някаква стойност преди това, след присвояването тя се актуализира с новата стойност, т.е. новата стойност замества старата.

### ПРИМЕР – Присвояване на стойност

`име = израз ;`



#### ПРИМЕР

```
String strGreeting;
strGreeting = name + ", привет!";
```

Ако променливата `name` има стойност „Рони“, то променливата `strGreeting` ще получи стойност „Рони, привет!“.

#### ЗАДАЧА

В някоя хотели при включване на телевизора се появява съобщение, което пожелава приятен престой на клиента.

Създайте приложение `Hotel`, което симулира работата на телевизора.

Полученото трябва да въвежда името на клиента и да изведе желанието.

## СТЪПКА ПО СТЪПКА – бутони

9. За всеки бутон създайте метод, асоцииран със събитието по подразбиране.
10. Задайте функционалност на бутона **Изчисти**: в текстовото поле и етикетите с поздрав трябва да има празен низ.
11. Задайте функционалност на бутона **Изход**.
12. Според анализа по-горе задайте функционалност на бутона **Запознай се**.

Контекстно меню: `Add event handler` → `Action` → `actionPerformed`

### Бутон Изчисти

```
btnReset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        txtName.setText("");
        lblGreeting1.setText("");
        lblGreeting2.setText("");
    }
});
```

### Бутон Изход

```
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
```

### Бутон Запознай се

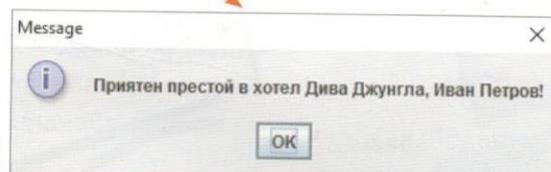
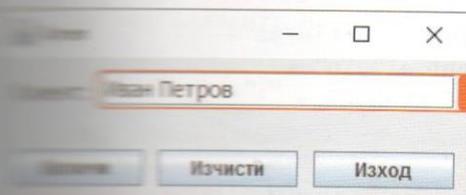
```
btnGreeting.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name = txtName.getText();

        String strGreeting1 = "Здравей, " + name + "!";
        lblGreeting1.setText(strGreeting1);

        String strGreeting2 = name + ", Java иска да работи с теб!";
        lblGreeting2.setText(strGreeting2);
    }
});
```

## УПЪТВАНЕ

За извеждане на съобщението използвайте стандартна диалогова кутия (`JOptionPane`).



**В ТЕМАТА:**

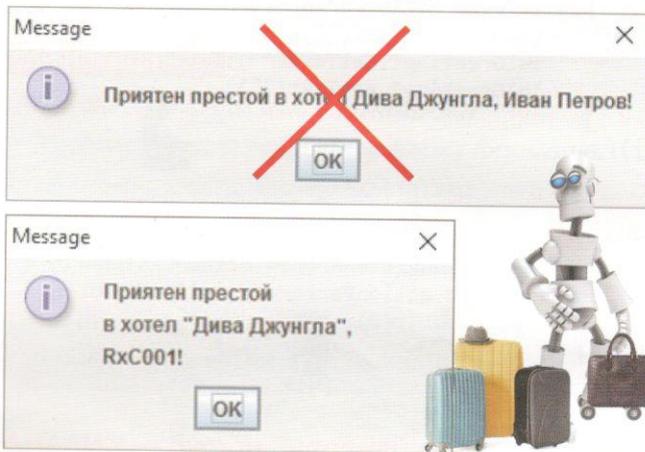
- ✓ Защо специалните символи са *специални*.
- ✓ Как се форматира низ за извеждане.
- ✓ Кога и как се съкращава записът на оператора за присвояване.

**ЗАДАЧА 1**

Съобщението, което се извежда от приложението **Hotel**, има някои недостатъци:

- Името на хотела би трябвало да е в кавички.
- Текстът е изписан на един ред. Ако стане прекалено дълъг, може да не се събере на екрана.

Коригирайте програмата така, че да избегнете недостатъците.



**СПЕЦИАЛНИ СИМВОЛИ**

Някои символи не може да се запишат директно в символен низ, тъй като имат специално предназначение – например символът `"`, или нямаат графичен образ – например знака за нов ред.

Тези символи се наричат **специални** и се изписват с двойка знаци без интервал между тях.

Някои специални символи	
<code>\n</code>	нов ред
<code>\t</code>	табулация
<code>\"</code>	кавички
<code>\\</code>	лява наклонена черта



**ПРИМЕР 1**

```
String client = txtClient.getText();
String strWelcome =
    "Приятен престой\n"
    + "в хотел \"Дива Джунгла\", \n"
    + client + "!";
```

**ИМЕНУВАНИ КОНСТАНТИ**

Въпреки че резултатът вече е задоволителен, добре е дългата константна част на съобщението да се отдели като самостоятелна **именувана константа**. Това ще позволи лесното ѝ многократно използване при нужда или бързо редактиране на цялото съобщение.

Константи се декларират с помощта на ключовата дума **final**, която забранява последваща промяна на стойността.

**ПРАВИЛО – Деклариране на константа**

```
final тип име = стойност;
```

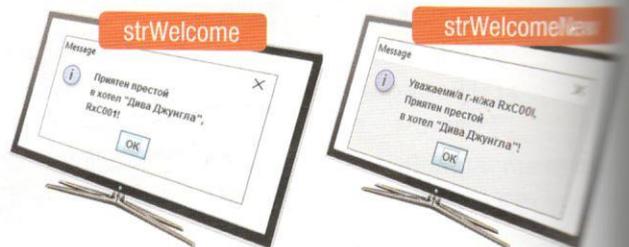
**ПРИМЕР 2**

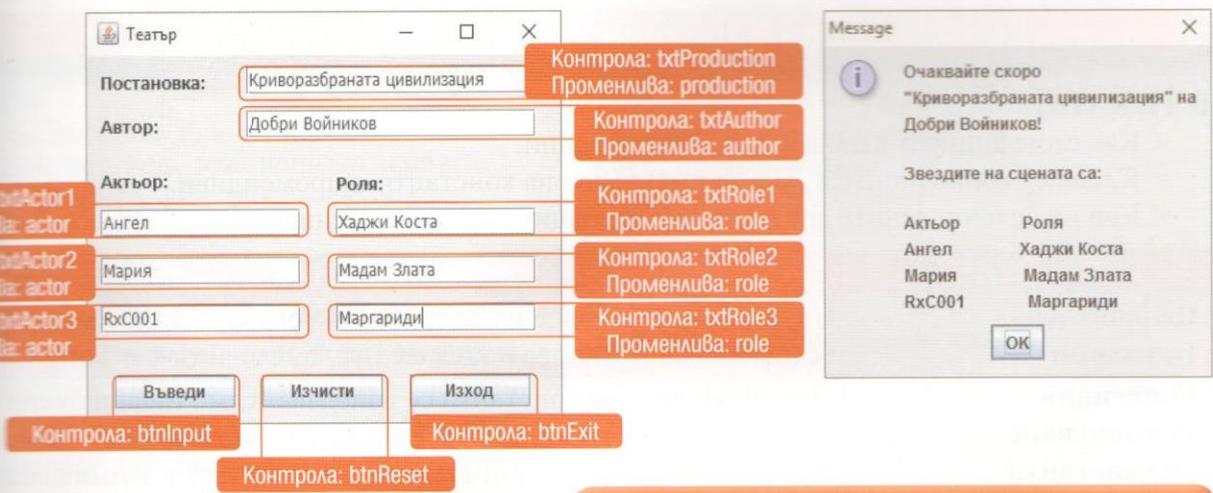
```
final String BASIC_MESSAGE =
    "Приятен престой\n"
    +"в хотел \"Дива Джунгла\"";
```

**ПРИМЕР 3**

```
String strWelcome =
    BASIC_MESSAGE + ", \n" +
    client + "!";
```

```
String strWelcomeNew =
    "Уважаеми/а г-н/жа +
    client + ", \n" +
    BASIC_MESSAGE + "!";
```





## ЗАДАЧА 2

Изградете приложение **Theatre**, което извежда рекламен афиш за постановка на училищния театър. Приложението трябва да въвежда името и автора на постановката и да извежда имените – тримата главни актьори и техните роли.

## УПЪТВАНЕ

Използвайте именуванни константи за съобщенията, които не се променят.

## ПРЕДИМО – Съкратен запис за присвояване

променлива += израз ;

Знаецава, че **променливата** се **увеличава** (категориално) с **израза**, т.е.

променлива = променлива + израз ;

## ПРИМЕР – начало на съобщението

```
String production = txtProduction.getText();
String author = txtAuthor.getText();
String message = "Очаквайте скоро \n\"
    + production + "\" на\n\"
    + author + "! \n\n";
message += "Звездите на сцената са:\n\n";
```

## ФОРМАТИРАНЕ НА НИЗ

Подреждането на данните в таблица може да стане чрез метода `String.format(str, ...)`. Резултатът е низ, оформен по определени правила.

## УПЪТВАНЕ

Използвайте съкратената форма на оператора за присвояване, за да добавите към съобщението данните за актьорите и ролите им.

## ПРИМЕР – оформяне на колони

```
String actor;
String role;
actor = txtActor1.getText();
role = txtRole1.getText();
message += String.format("%-20s %-20s\n",
    actor, role);
```

## ЗАДАЧА 2А

Променете приложението **Theatre** така, че при стартиране в текстовите полета да се извеждат подсказки относно това, което трябва да се въведе.

