|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| Кафедра КБ-14 «Цифровые технологии обработки данных» |

**ОТЧЕТ**

**по практической работе №2**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил |  |  | Бурмистров И.Г.  *фамилия, имя, отчество* |
| шифр | 22Б0616 | группа | БСБО-07-22 |
|  |  |  |  |
| Проверил |  | *ученая степень, должность* | Изергин Д.А.  *фамилия, имя, отчество* |

**Москва 2025г.**

В ходе данной работы было созданы следующие модули: «activitylifecycle», «multiactivity», «intentfilter», «toastapp», «notoficationapp», «dialog» (см. Рисунок 1).

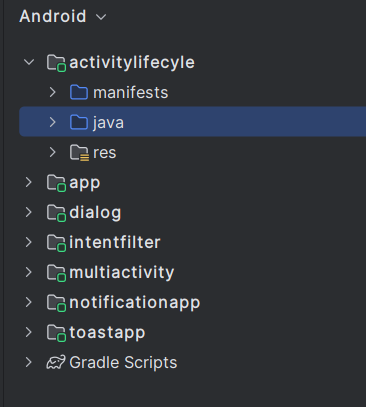


Рисунок 1. Созданные модули

В модуле «activitylifecycle» были переопределены основные методы жизненного цикла родительского класса для просмотра жизненного цикла активности (см. Листинг 1).

public class MainActivity extends AppCompatActivity {  
  
 private static final String *TAG* = "ActivityLifeCycle";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Log.*d*(*TAG*, "метод OnCreate сработал!");  
 }  
  
 @Override  
 protected void onStart(){  
 super.onStart();  
 Log.*d*(*TAG*, "метод OnStart сработал!");  
 }  
  
 @Override  
 protected void onResume(){  
 super.onResume();  
 Log.*d*(*TAG*, "Метод onResume Сработал!");  
 }  
  
 @Override  
 protected void onStop(){  
 super.onStop();  
 Log.*d*(*TAG*, "Метод onStop Сработал!");  
 }  
  
 @Override  
 protected void onRestart(){  
 super.onRestart();  
 Log.*d*(*TAG*, "Метод onRestart Сработал!");  
 }  
  
 @Override  
 protected void onDestroy(){  
 super.onDestroy();  
 Log.*d*(*TAG*, "метод onDestroy Сработал!");  
 }

}

Листинг 1. Переопределённые методы

После запуска приложения в «logcat» были просмотрены отработанные методы (см. Рисунок 2).

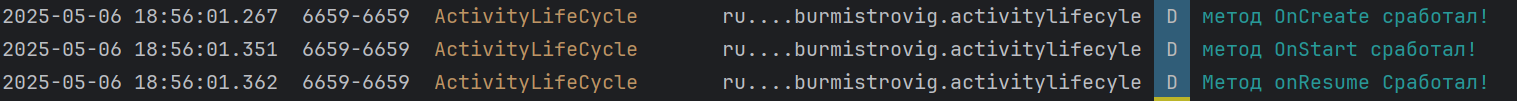
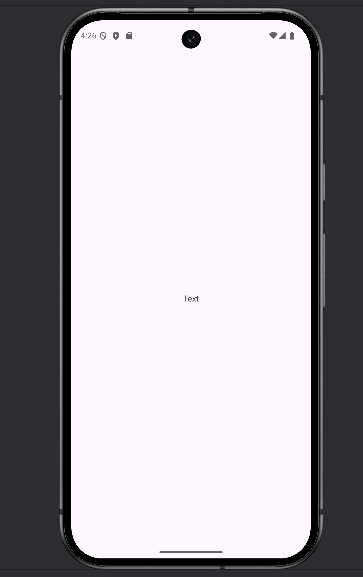
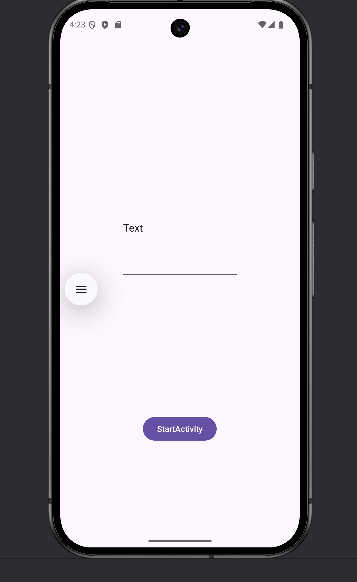


Рисунок 2. Вызванные методы

В модуле «MultiActivityh» был реализован переход от одной активности в другую при помощи кнопок и фильтра «intent». После ввода данных и нажатия на кнопку «Отправить» открывается другая активность, в которой видно введённый текст. При переходе на новую активность в ней вызываются те же методы, что и при открытии первой (см. Рисунок 3).



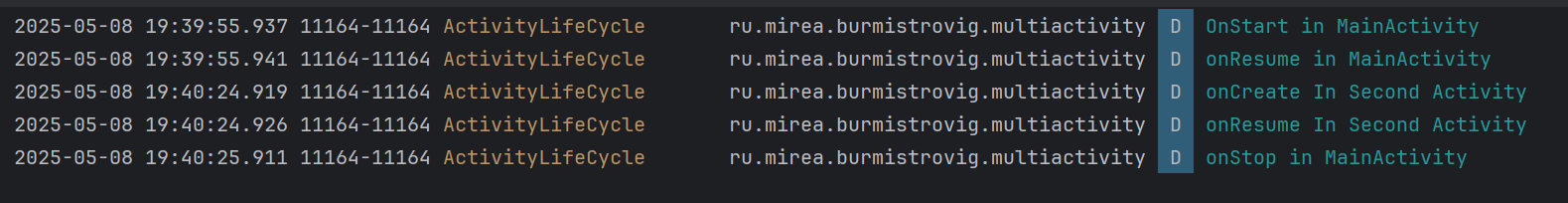


Рисунок 3. Пример работы приложения

Передача текста с поля производится с помощью метода «putExtra». Затем во второй активности возвращается значение через метод «getIntent» и передается в метод «setText» (см. Листинг 2).

String resieveddata = (String) getIntent().getSerializableExtra("key");  
  
TextView textSecond = findViewById(R.id.*textSecond*);  
  
textSecond.setText(resieveddata);  
  
Log.*d*(*TAG*, "onCreate In Second Activity");

Листинг 2. Передача текста

Также был создан модуль «IntentFilter» (см. рис. 4).

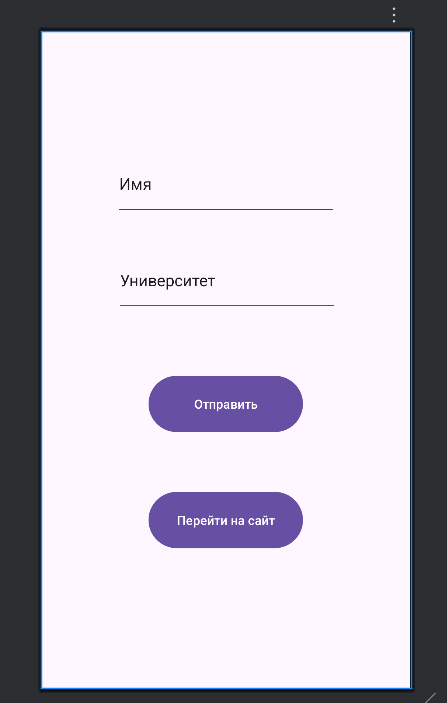


Рисунок 4. Модуль «IntentFilter»

В нём была создана кнопка, перенаправляющая на сайт «РТУ МИРЭА» (см. листинг 3).

public void OnClickURL(View view){  
 Uri address = Uri.*parse*("https://www.mirea.ru/");  
 Intent openLink = new Intent(Intent.*ACTION\_VIEW*, address);  
 startActivity(openLink);  
}

Листинг 3. Перенаправление по ссылке

Затем была добавлена кнопка «Отправить», которая открывает окно «Поделиться текстом» (см. Рисунок 5).

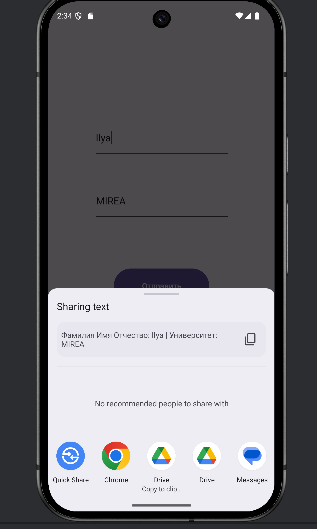
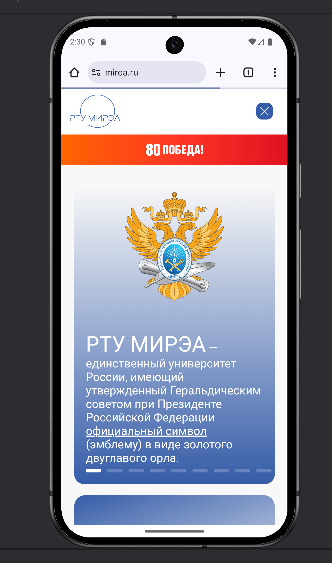
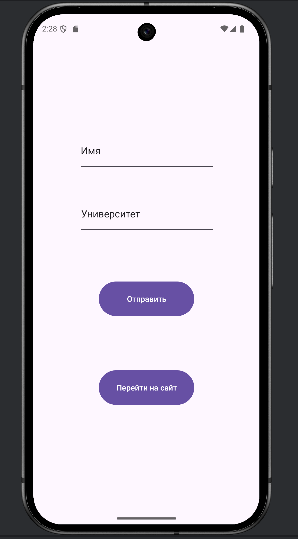
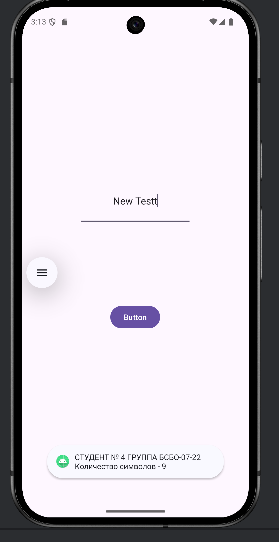


Рисунок 5. Пример работы приложения

Далее был создано модулей «ToatApp», в котором было реализовано уведомление, которое вызывается по кнопке и показывает сколько символов введено в тестовое поле (см. Рисунок 5).

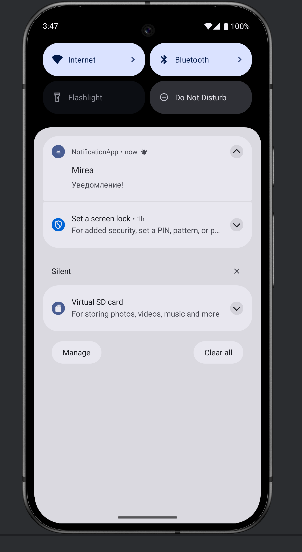
  
Рисунок 5. Пример работы модуля

Код данного модуля представлен в листинге 4.

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
  
 }  
 public void OnClickToast(View view){  
 EditText text = findViewById(R.id.*editTextText*);  
 Toast toast = Toast.*makeText*(getApplicationContext(),  
 "СТУДЕНТ № 4 ГРУППА БСБО-07-22 Количество символов - " + text.getText().length(),  
 Toast.*LENGTH\_SHORT*);  
 toast.show();  
 }  
}

Листинг 4. Уведомление с подсчётом символов

Далее был создан ещё один модуль «NotificationApp, также вызывающий уведомление при нажатии на кнопку, но уже в разделе уведомлений от приложений (см. Рисунок 6).

  
Рисунок 6. Уведомление

Код данного модуля представлен в листинге 5.

public class MainActivity extends AppCompatActivity {  
 private static final String *CHANNEL\_ID* = "com.mirea.asd.notification.ANDROID";  
 private int PermissionCode = 200;  
 @RequiresApi(api = Build.VERSION\_CODES.*TIRAMISU*)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
  
 if (ContextCompat.*checkSelfPermission*(this, *POST\_NOTIFICATIONS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 Log.*d*(MainActivity.class.getSimpleName().toString(), "Разрешения получены");  
 } else {  
 Log.*d*(MainActivity.class.getSimpleName().toString(), "Нет разрешений!");  
 ActivityCompat.*requestPermissions*(this, new String[]{*POST\_NOTIFICATIONS*}, PermissionCode);  
 }  
 }  
 public void OnClickSendNotification(View view) {  
 if (ActivityCompat.*checkSelfPermission*(this, *POST\_NOTIFICATIONS*) != PackageManager.*PERMISSION\_GRANTED*) {  
 return;  
 }  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setContentText("Congratulation!")  
 .setSmallIcon(R.drawable.*ic\_launcher\_foreground*)  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setStyle(new NotificationCompat.BigTextStyle()  
 .bigText("Уведомление!"))  
 .setContentTitle("Mirea");  
  
 int importance = NotificationManager.*IMPORTANCE\_DEFAULT*;  
 NotificationChannel channel = null;  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 channel = new NotificationChannel(*CHANNEL\_ID*, "Student FIO Notification", importance);  
 }  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 channel.setDescription("MIREA Channel");  
 }  
  
 NotificationManagerCompat notificationManager = NotificationManagerCompat.*from*(this);  
 notificationManager.createNotificationChannel(channel);  
 notificationManager.notify(1, builder.build());  
 }  
}

Листинг 5. Настройка уведомления

Наконец был создан модуль «Dialog». В данном модуле есть текстовое поле с датой и временем, а также 4 кнопки: первая из них показывает уведомление, вторая и третья меняют время и дату в тестовом поле, а последняя показывает всплывающее окно загрузки (см. Рисунок 7).

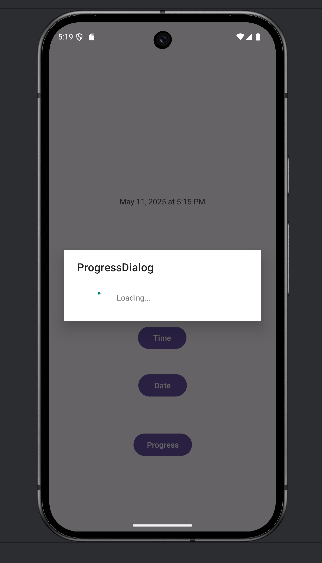
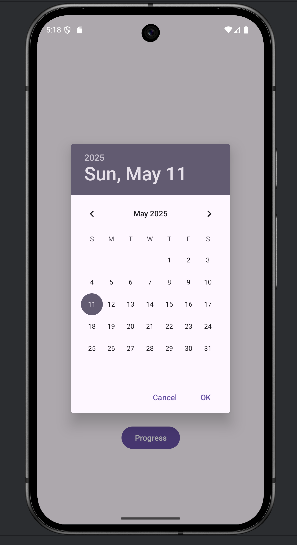
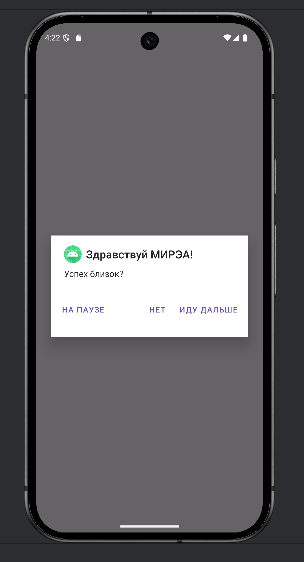
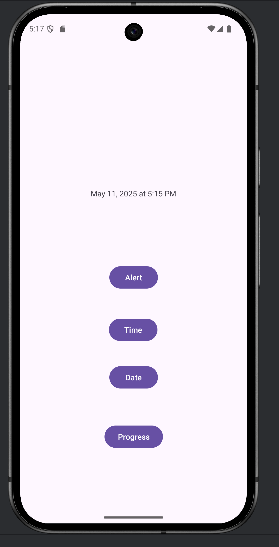


Рисунок 7. Модуль «Dialog»

Код файла «MainActivity» представлен в листинге 6.

public class MainActivity extends AppCompatActivity {  
  
 TextView currentDateTime;  
 Calendar dateAndTime=Calendar.*getInstance*();  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
  
 currentDateTime = findViewById(R.id.*textView*);  
 setInitialDateTime();  
 }  
 public void OnClickShowDialog(View view) {  
 AlertDialogFragment alertDialog = new AlertDialogFragment();  
 alertDialog.show(getSupportFragmentManager(), "mirea");  
 }  
  
 public void onOkClicked() {  
 Toast.*makeText*(getApplicationContext(), "Вы выбрали кнопку \"возможно\"!",  
 Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void onCancelClicked() {  
 Toast.*makeText*(getApplicationContext(), "Вы выбрали кнопку \"нет\"!",  
 Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void onNeutralClicked() {  
 Toast.*makeText*(getApplicationContext(), "Вы выбрали кнопку \"вроде нет\"!",  
 Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void OnClickTimeDialog(View view) {  
  
 TimeClass time = new TimeClass(MainActivity.this, dateAndTime,  
 (timePicker, hourOfDay, minute) ->{  
 dateAndTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);  
 dateAndTime.set(Calendar.*MINUTE*, minute);  
 setInitialDateTime();  
 });  
 time.show();  
 }  
  
 private void setInitialDateTime() {  
  
 currentDateTime.setText(DateUtils.*formatDateTime*(this,  
 dateAndTime.getTimeInMillis(),  
 DateUtils.*FORMAT\_SHOW\_DATE* | DateUtils.*FORMAT\_SHOW\_YEAR* | DateUtils.*FORMAT\_SHOW\_TIME*));  
 }  
  
 public void OnClickDate(View view)  
 {  
 new DateClass(MainActivity.this, dateAndTime,  
 (datePicker, year, monthOfYear, dayOfMonth) -> {  
 dateAndTime.set(Calendar.*YEAR*, year);  
 dateAndTime.set(Calendar.*MONTH*, monthOfYear);  
 dateAndTime.set(Calendar.*DAY\_OF\_MONTH*, dayOfMonth);  
 setInitialDateTime();  
 }).show();  
 }  
  
 public void OnClickProgress(View view)  
 {  
  
 ProgressClass progress = new ProgressClass(this);  
 progress.show();  
  
 new Handler().postDelayed(()-> progress.dismiss(), 3000);  
 }  
}

Листинг 7. «MainActivity»

В листинге 8 представлен класс, отвечающий за вывод уведомления с вариантами ответов.

public class AlertDialogFragment extends DialogFragment {  
 @NonNull  
 @Override  
 public Dialog onCreateDialog(Bundle savedInstanceState) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
 builder.setTitle("Здравствуй МИРЭА!")  
 .setMessage("Успех близок?")  
 .setIcon(R.mipmap.*ic\_launcher\_round*)  
 .setPositiveButton("Иду дальше", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 ((MainActivity)getActivity()).onOkClicked();  
 dialog.cancel();  
  
 }  
 })  
 .setNeutralButton("На паузе", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 ((MainActivity)getActivity()).onNeutralClicked();  
 dialog.cancel();  
 }  
 })  
 .setNegativeButton("Нет", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 ((MainActivity)getActivity()).onCancelClicked();  
 dialog.cancel();  
 }  
 });  
 return builder.create();  
 }  
}

Листинг 8. Класс «AlertDialogFragment»

В листингах 9 и 10 соответственно представлены классы, отвечающие за изменение даты и времени в текстовом поле.

public DateClass(Context context, Calendar initialDate, OnDateSetListener listener) {  
 super(context, (datePicker, year, month, day) -> {  
 if (listener != null) {  
 listener.onDateSet(datePicker, year, month, day);  
 initialDate.set(Calendar.*YEAR*, year);  
 initialDate.set(Calendar.*MONTH*, month);  
 initialDate.set(Calendar.*DAY\_OF\_MONTH*, day);  
 }  
 },  
 initialDate.get(Calendar.*YEAR*),  
 initialDate.get(Calendar.*MONTH*),  
 initialDate.get(Calendar.*DAY\_OF\_MONTH*));  
  
 this.initialDate = initialDate;  
}

Листинг 9. Класс «DateClass»

public class TimeClass extends TimePickerDialog {  
 private final Calendar initialTime;  
 private final OnTimeSetListener externalListener;  
  
 public TimeClass(Context context, Calendar initialTime, OnTimeSetListener listener) {  
 super(context, (TimePicker, hourOfDay, minute) -> {  
 if (listener != null) {  
 listener.onTimeSet(TimePicker, hourOfDay, minute);  
 initialTime.set(Calendar.*HOUR\_OF\_DAY*, hourOfDay);  
 initialTime.set(Calendar.*MINUTE*, minute);  
 }  
 },  
 initialTime.get(Calendar.*HOUR\_OF\_DAY*),  
 initialTime.get(Calendar.*MINUTE*),  
 DateFormat.*is24HourFormat*(context));  
  
 this.initialTime = initialTime;  
 this.externalListener = listener;  
 }  
  
 @Override  
 public void onTimeChanged(TimePicker timePicker, int hourOfDay, int minute) {  
 super.onTimeChanged(timePicker, hourOfDay, minute);  
 }  
}

Листинг 9. Класс «TimeClass»

В листинге 10 представлен класс, отвечающий за вывод окна загрузки.

public class ProgressClass extends ProgressDialog {  
  
 public ProgressClass(Context context) {  
 super(context);  
 configureDialog();  
 }  
  
 public ProgressClass(Context context, int theme) {  
 super(context, theme);  
 configureDialog();  
 }  
  
 private void configureDialog() {  
 setTitle("ProgressDialog");  
 setMessage("Loading...");  
 setCancelable(false);  
  
 }  
  
  
 public void showWithSafetyCheck() {  
 if (!isShowing()) {  
 show();  
 }  
 }  
  
 @Override  
 public void dismiss() {  
 if (isShowing()) {  
 super.dismiss();  
 }  
 }  
}

Листинг 10. Класс «ProgressClass»