N° d'inscription : 44561

APPLICATION ANDROID ENCOURAGEANT L'ACTIVITÉ SPORTIVE PAR LIMITATION DU TEMPS D'ÉCRAN

Comment empêcher une utilisation chronophage du smartphone tout en réinvestissant ce temps en activité sportive ?

RAGOT Cyrian

SOMMAIRE

Introduction au projet

- Pourquoi ce projet ?
- Objectifs

L'application

- Présentation générale
- Structure technique de l'application
- Mesure de l'effort physique
- Un exemple de fonction

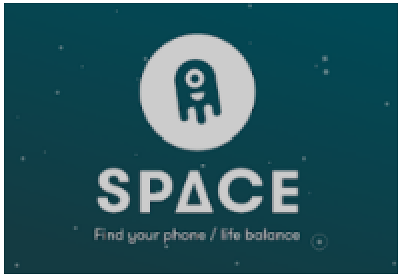
Le calcul de vitesse

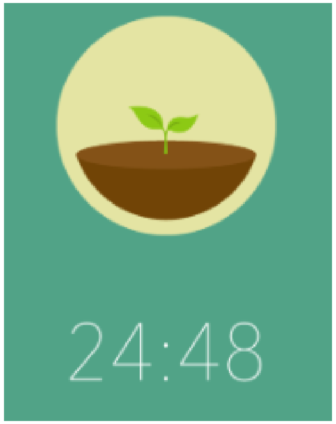
- Modèles de calcul
- Protocole expérimental
- Résultats
- Conclusion et modèle retenu

Conclusion

- Exigences
- Difficultés rencontrées
- Améliorations possibles
- Retour à la problématique

Comment ne pas être accro à son smartphone ? (et pourquoi l'est-on ?)





Pourquoi ce projet ?

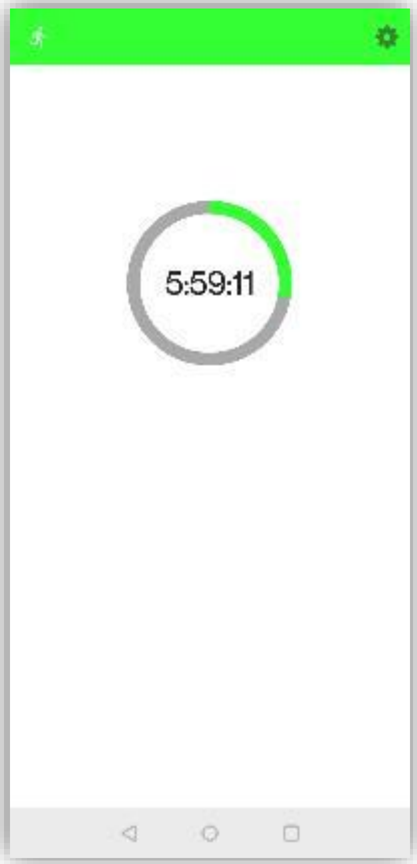
→ Problème personnel
→ Addiction au smartphone
→ Systèmes de récompense

→ Innovation unique
→ Opportunité personnelle

→ Ancrage dans le thème

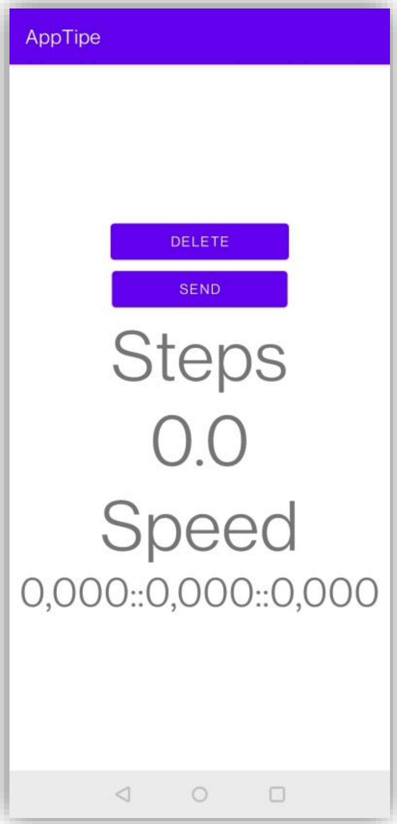
Rèfs : Cyrus North (Youtube), Temps Présent (Youtube), Space (Application), Forest (Application)

L'application android



5:59:11

L'évolution du développement

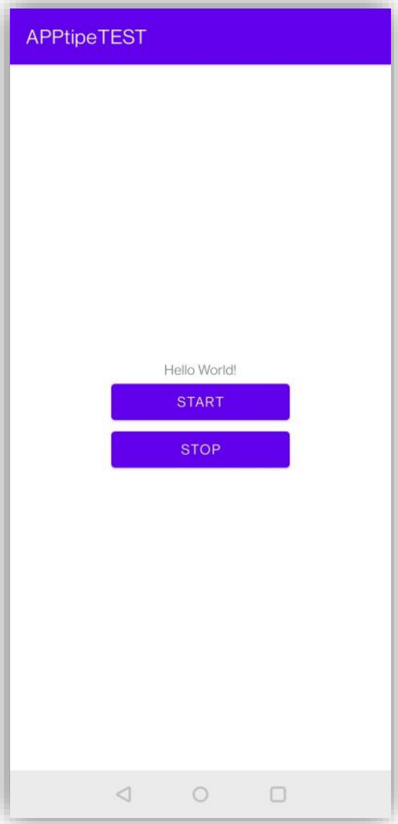

AppTipe

Steps

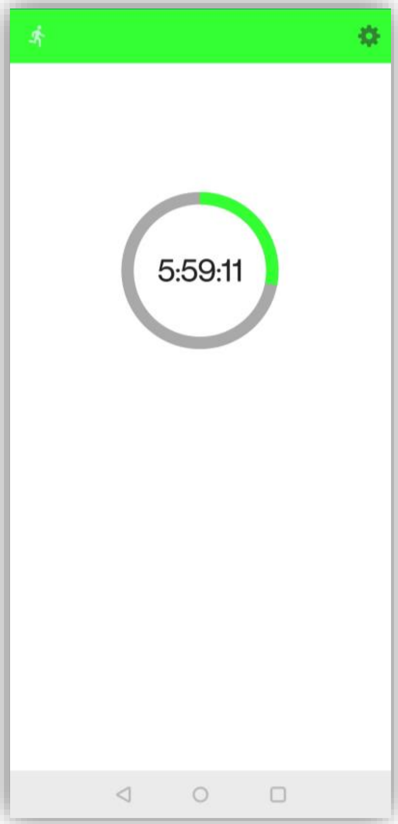
0.0

Speed

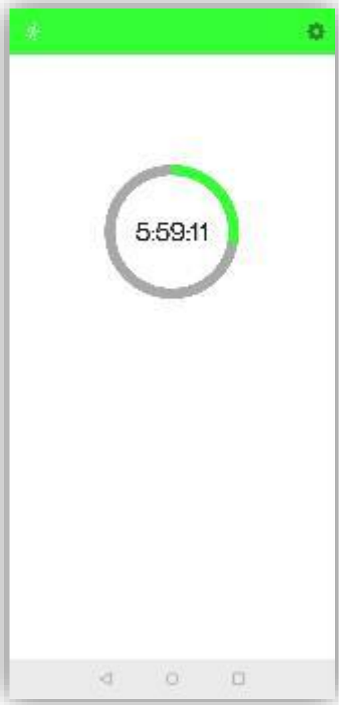
0,000::0,000::0,000

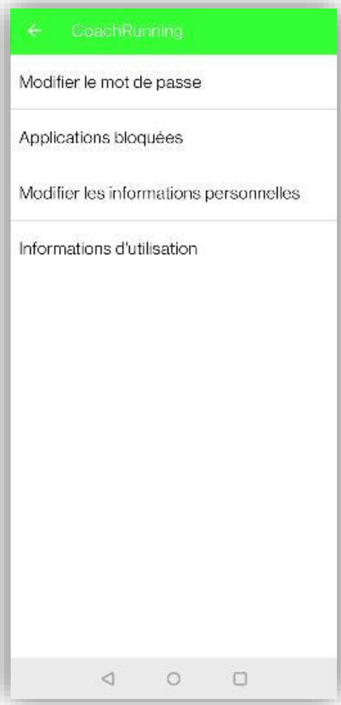

APPtipeTEST

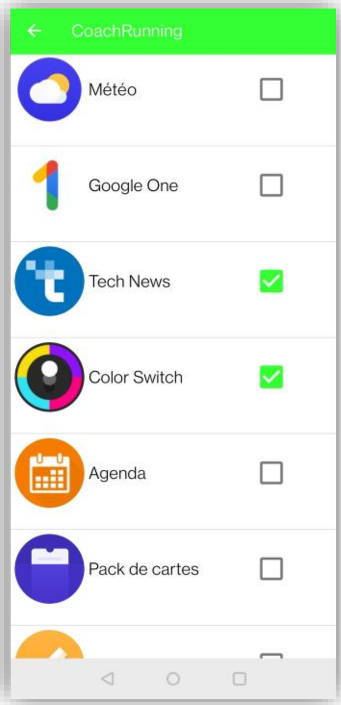
Hello World!


5:59:11

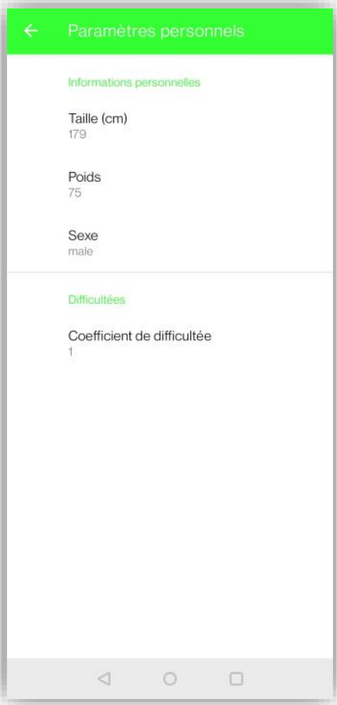
Présentation générale de l'application




CoachRunning
Modifier le mot de passe
Applications bloquées
Modifier les informations personnelles
Informations d'utilisation


CoachRunning

- Météo □
- Google One □
- Tech News ☑
- Color Switch ☑
- Agenda □
- Pack de cartes □


Paramètres personnels

Informations personnelles

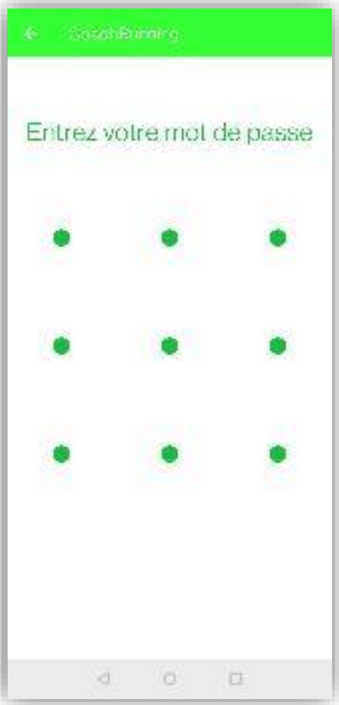
Taille (cm)
179

Poids
75

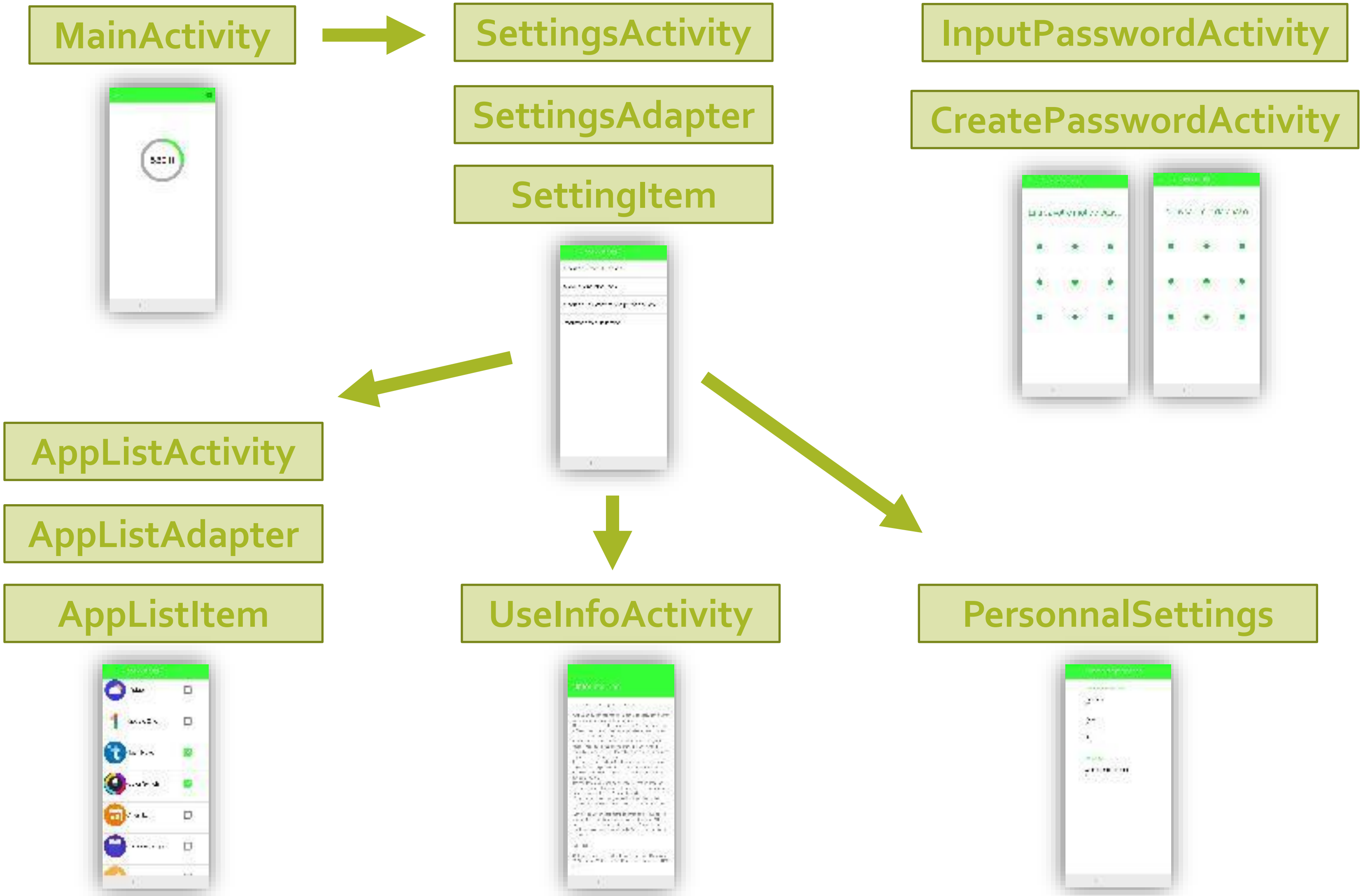
Sexe
male

Difficultées

Coefficient de difficultée
1



Structure du code



MainActivity

SettingsActivity

SettingsAdapter

SettingItem

InputPasswordActivity

CreatePasswordActivity

AppListActivity

AppListAdapter

AppListItem

UseInfoActivity

PersonnalSettings

Structure du code

Autres classes :

StartMyServiceAtBootReceiver

MyApplication

DataFile

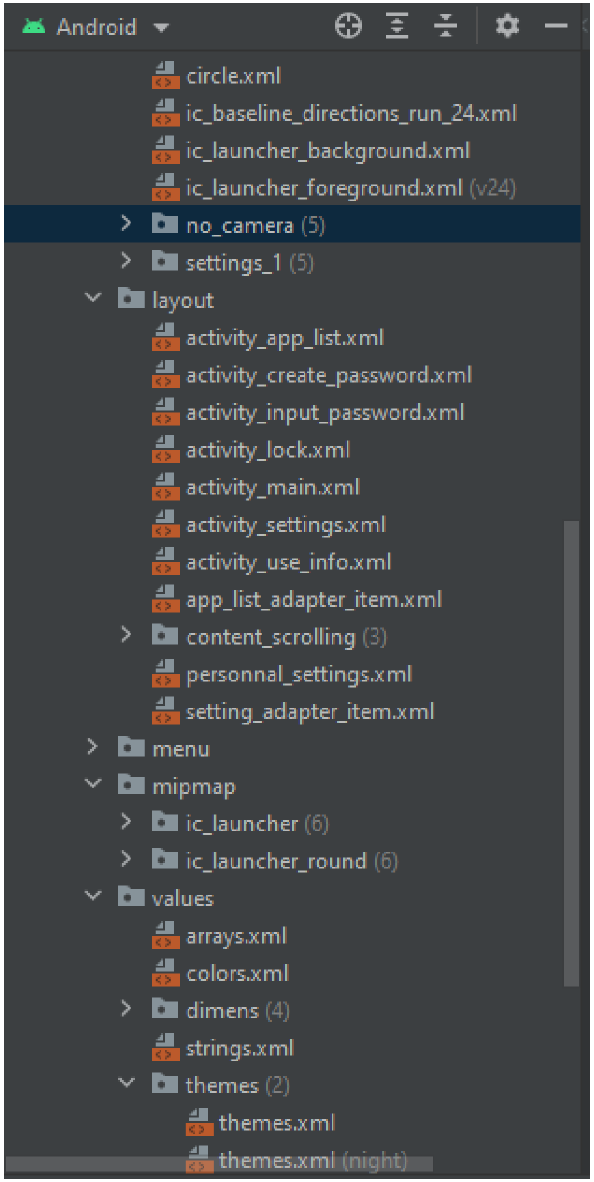
AndroidUtils

Interface :

MyActionCallback

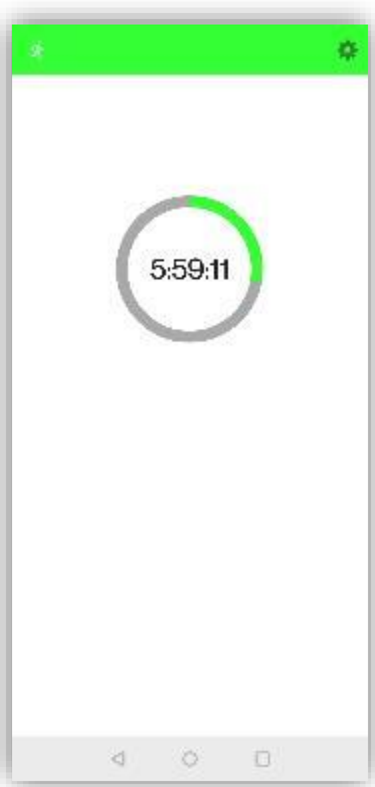
Service :

LockService



```
        circle.xml
        ic_baseline_directions_run_24.xml
        ic_launcher_background.xml
        ic_launcher_foreground.xml (v24)
    >   no_camera (5)
    >   settings_1 (5)
  v   layout
        activity_app_list.xml
        activity_create_password.xml
        activity_input_password.xml
        activity_lock.xml
        activity_main.xml
        activity_settings.xml
        activity_use_info.xml
        app_list_adapter_item.xml
    >   content_scrolling (3)
        personnal_settings.xml
        setting_adapter_item.xml
  >   menu
  v   mipmap
    >   ic_launcher (6)
    >   ic_launcher_round (6)
  v   values
        arrays.xml
        colors.xml
    >   dimens (4)
        strings.xml
    v   themes (2)
          themes.xml
          themes.xml (night)
```

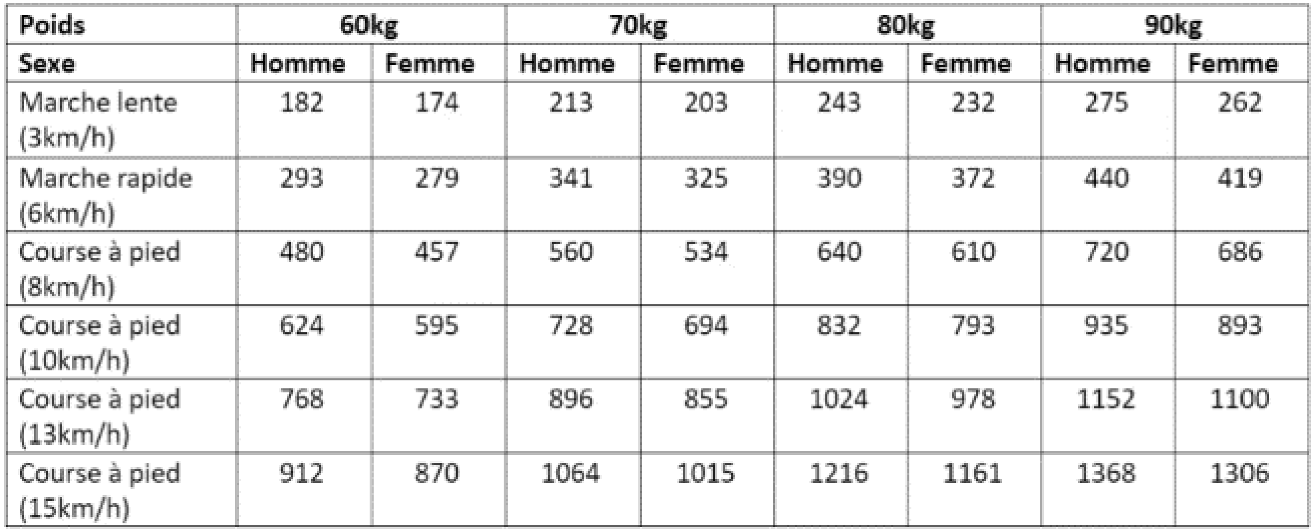
Mesure de l'effort physique



Exemples de coefficients de difficulté :

→ **1** : 1h de course à 8 km/h

→ **0.17** : marathon à 8 km/h

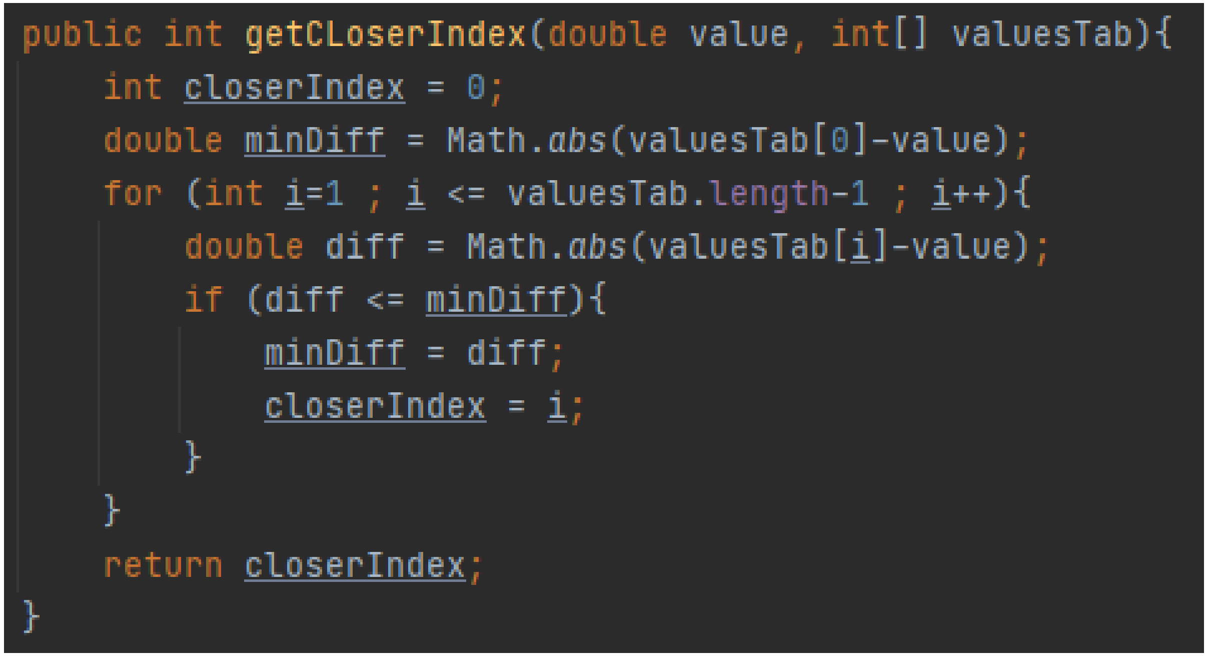


Poids	60kg		70kg		80kg		90kg	
Sexe	Homme	Femme	Homme	Femme	Homme	Femme	Homme	Femme
Marche lente (3km/h)	182	174	213	203	243	232	275	262
Marche rapide (6km/h)	293	279	341	325	390	372	440	419
Course à pied (8km/h)	480	457	560	534	640	610	720	686
Course à pied (10km/h)	624	595	728	694	832	793	935	893
Course à pied (13km/h)	768	733	896	855	1024	978	1152	1100
Course à pied (15km/h)	912	870	1064	1015	1216	1161	1368	1306

Estimation des dépenses caloriques en course à pied pour 1h d'effort

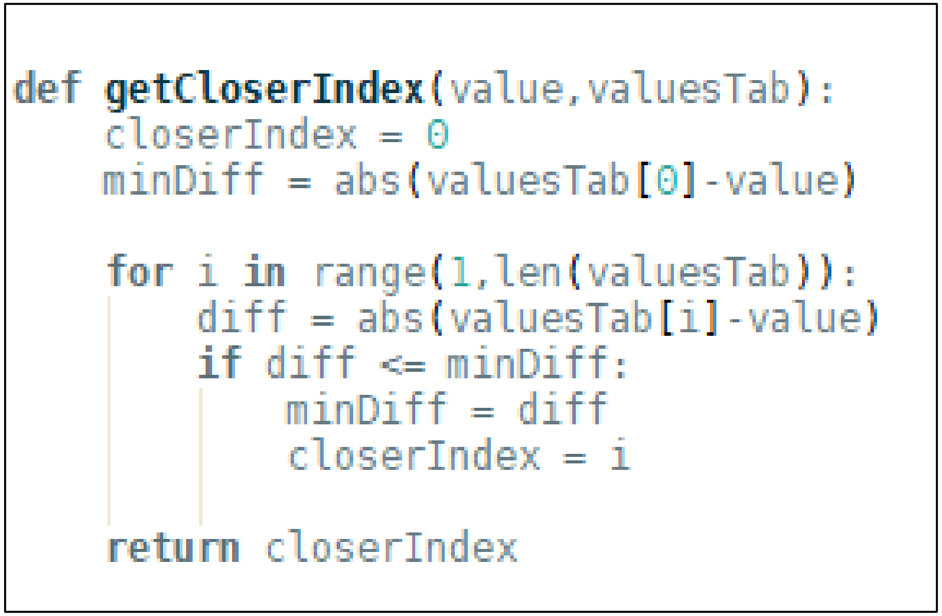
Kalenji.fr

Un exemple de fonction



```java
public int getCLoserIndex(double value, int[] valuesTab){
    int closerIndex = 0;
    double minDiff = Math.abs(valuesTab[0]-value);
    for (int i=1 ; i <= valuesTab.length-1 ; i++){
        double diff = Math.abs(valuesTab[i]-value);
        if (diff <= minDiff){
            minDiff = diff;
            closerIndex = i;
        }
    }
    return closerIndex;
}
```

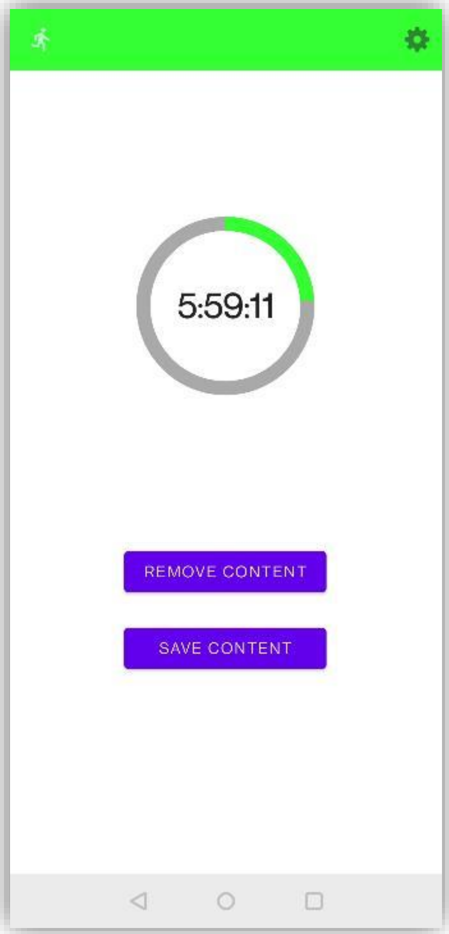
JAVA



```python
def getCloserIndex(value,valuesTab):
    closerIndex = 0
    minDiff = abs(valuesTab[0]-value)

    for i in range(1,len(valuesTab)):
        diff = abs(valuesTab[i]-value)
        if diff <= minDiff:
            minDiff = diff
            closerIndex = i

    return closerIndex
```

PYTHON

Le calcul de vitesse



Modèles de calcul de la vitesse

$$v = \frac{\Delta S \ \times \ L}{10} \times \frac{3600}{10^5}$$

ΔS : $nombre\ de\ pas\ enregistrés\ durant\ les\ 10\ s$

L : $longueur\ d'un\ pas\ (cm)$

v : $vitesse\ calculée\ sur\ 10s\ (km.h^{-1})$

Modèles de calcul de la vitesse

Modèle 1 : **L proportionnel à la taille**

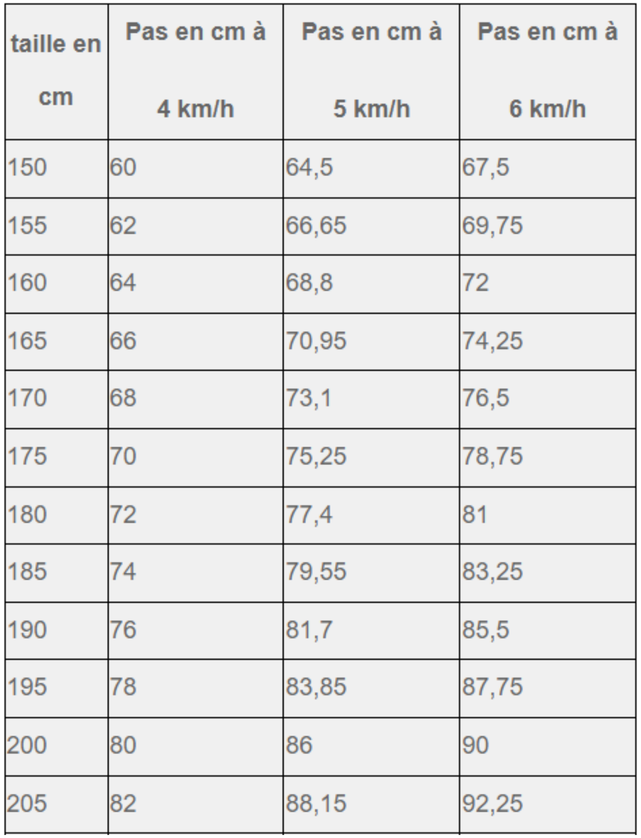
$$L = H \times \alpha$$

$$\alpha = 0.415\ (hommes)^{*}$$
$$\alpha = 0.413\ (femmes)$$

Modèle 2 : **L fonction linéaire de la vitesse**

$$L = 4.5 \times v + 54.3$$

(coefficient de corrélation : 0.99)



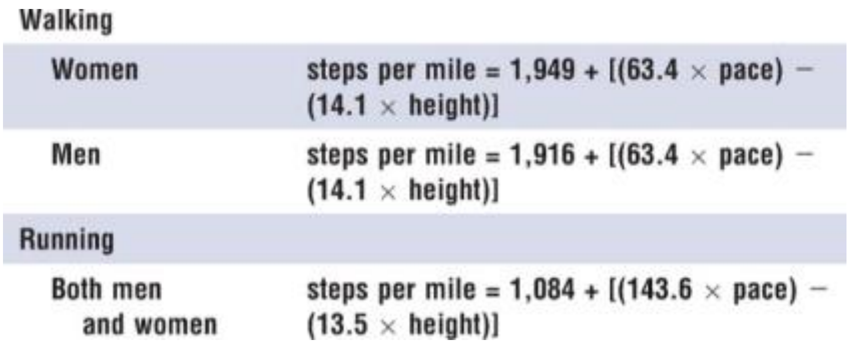
taille en cm	Pas en cm à 4 km/h	Pas en cm à 5 km/h	Pas en cm à 6 km/h
150	60	64,5	67,5
155	62	66,65	69,75
160	64	68,8	72
165	66	70,95	74,25
170	68	73,1	76,5
175	70	75,25	78,75
180	72	77,4	81
185	74	79,55	83,25
190	76	81,7	85,5
195	78	83,85	87,75
200	80	86	90
205	82	88,15	92,25

**

* d'après forum TomTom
** d'après objectifpleinair.com

Modèles de calcul de la vitesse

Modèle 3 : L fonction de la vitesse et la taille

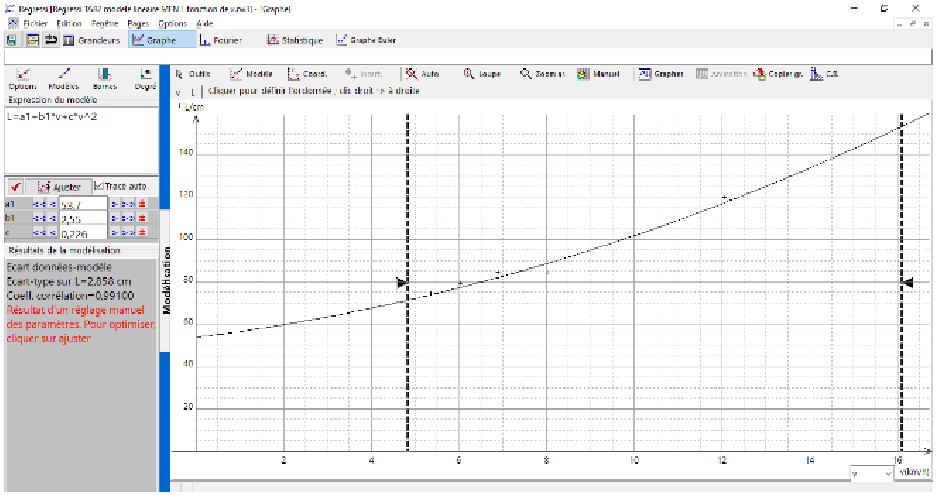


Walking	
Women	steps per mile = 1,949 + [(63.4 × pace) − (14.1 × height)]
Men	steps per mile = 1,916 + [(63.4 × pace) − (14.1 × height)]
Running	
Both men and women	steps per mile = 1,084 + [(143.6 × pace) − (13.5 × height)]

*

-pace en min/mile
-height en inches

Autres modèles :



Regressi [Regressi TVB2 modèle linéaire MEN l fonction de v.rw3] - [Graphe]

Expression du modèle

L=a1+b1*v+c*v^2

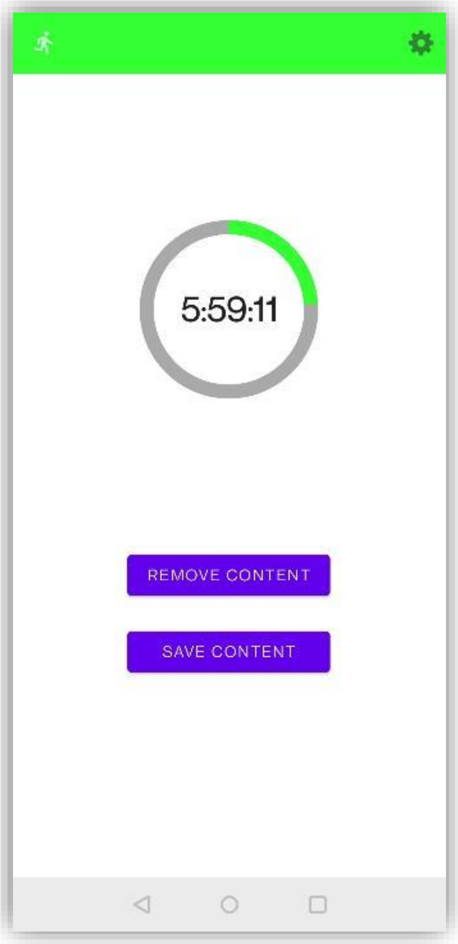
a1	53,7
b1	2,55
c	0,226

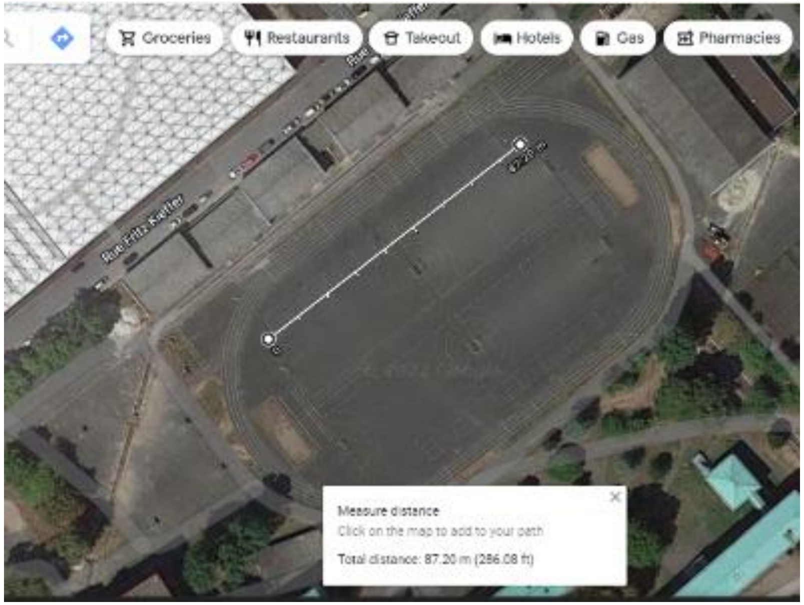
Résultats de la modélisation

Ecart données-modèle
Ecart-type sur L=2,858 cm
Coeff. corrélation=0,99100
Résultat d'un réglage manuel des paramètres. Pour optimiser, cliquer sur ajuster

* d’après un article du ACSM's Health & Fitness Journal

Protocole expérimental



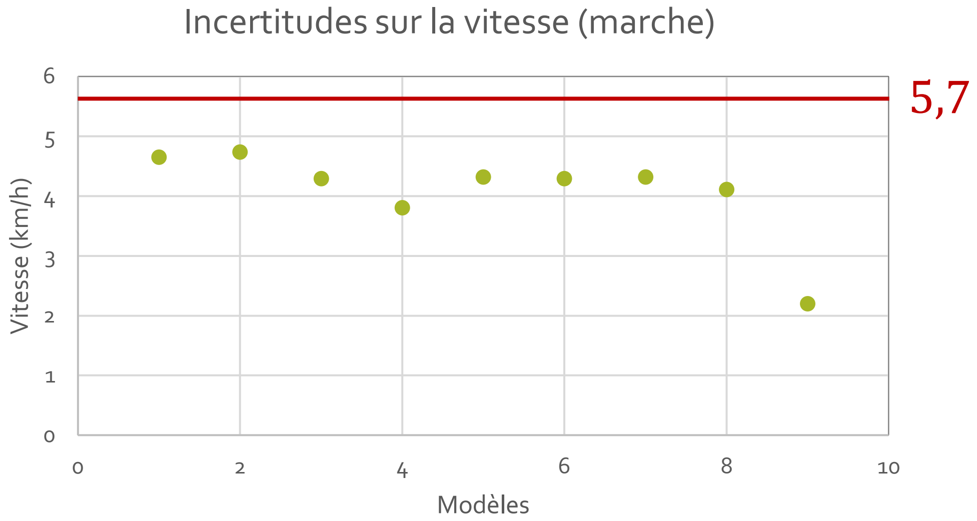


Measure distance

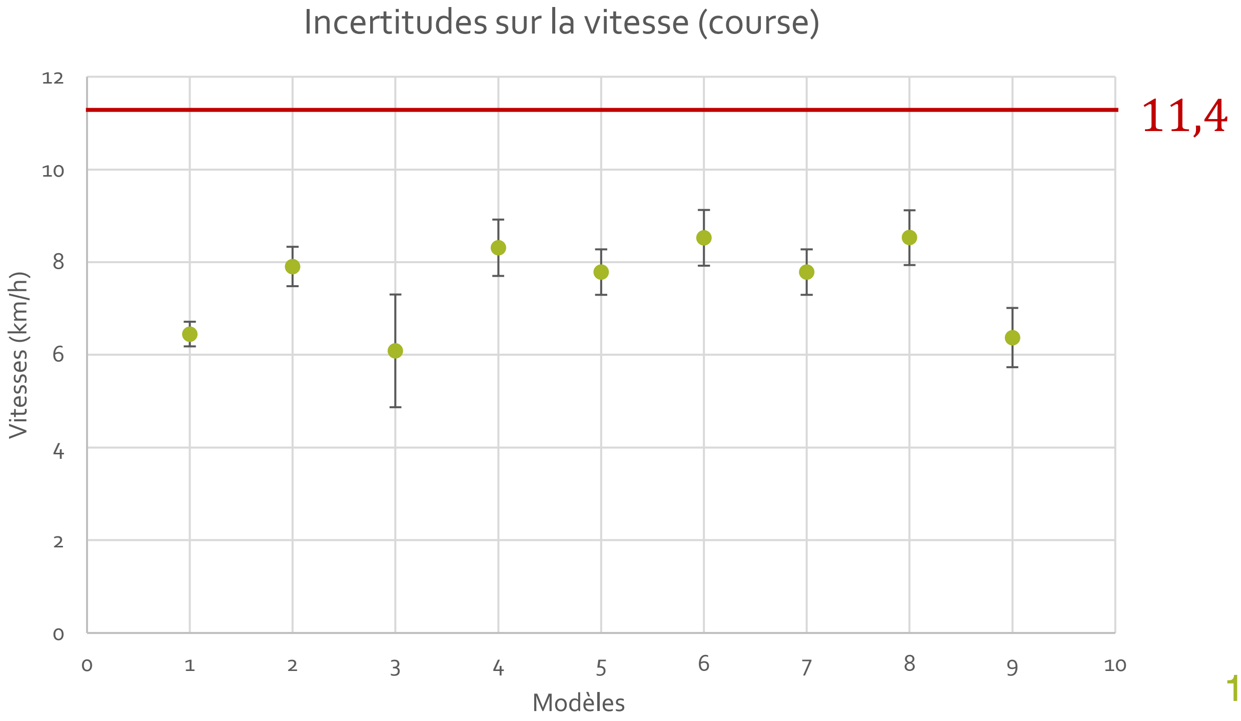
Click on the map to add to your path

Total distance: 87.20 m (286.08 ft)

- entrer les données de l'expérimentateur dans l'application (taille, sexe)
- se placer à un sommet du terrain
- appuyer REMOVE CONTENT
- lancer un chronomètre
- courir à une allure constante autour du terrain
- s'arrêter au même sommet après quelque tours
- stopper le chronomètre
- appuyer SAVE CONTENT


Incertitudes sur la vitesse (marche)
5,7
Vitesse (km/h)
0
1
2
3
4
5
6
0
2
4
6
8
10
Modèles

Résultats


Incertitudes sur la vitesse (course)
11,4
Vitesses (km/h)
0
2
4
6
8
10
12
0
1
2
3
4
5
6
7
8
9
10
Modèles

CONCLUSION

→ DES EXIGENCES PARTIELLEMENT SATISFAITES

→ DES DIFFICULTÉS RENCONTRÉES

→ DES AMÉLIORATIONS POSSIBLES

Comment empêcher une utilisation chronophage du smartphone tout en réinvestissant ce temps en activité sportive ?

Merci pour votre attention

CoachRunning

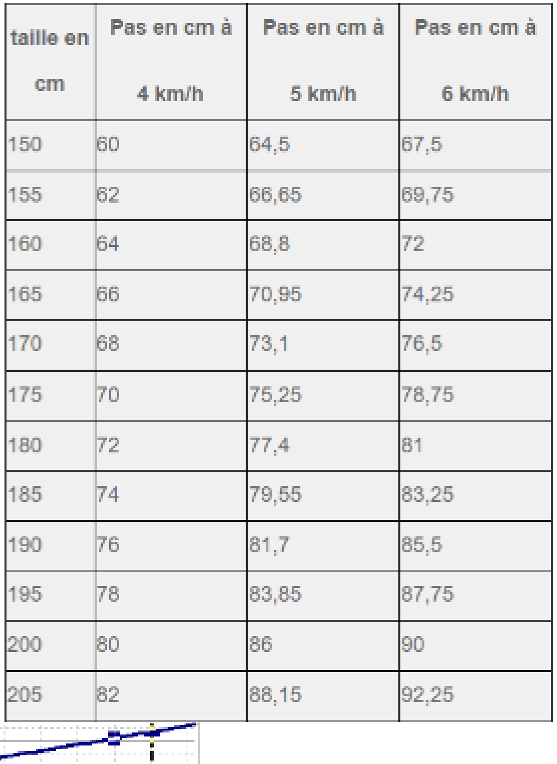
ANNEXES

Modèle 1

L proportionnel à la taille : $L = H \times \alpha$

$$\alpha = 0.415\ (hommes)$$
$$\alpha = 0.413\ (femmes)$$

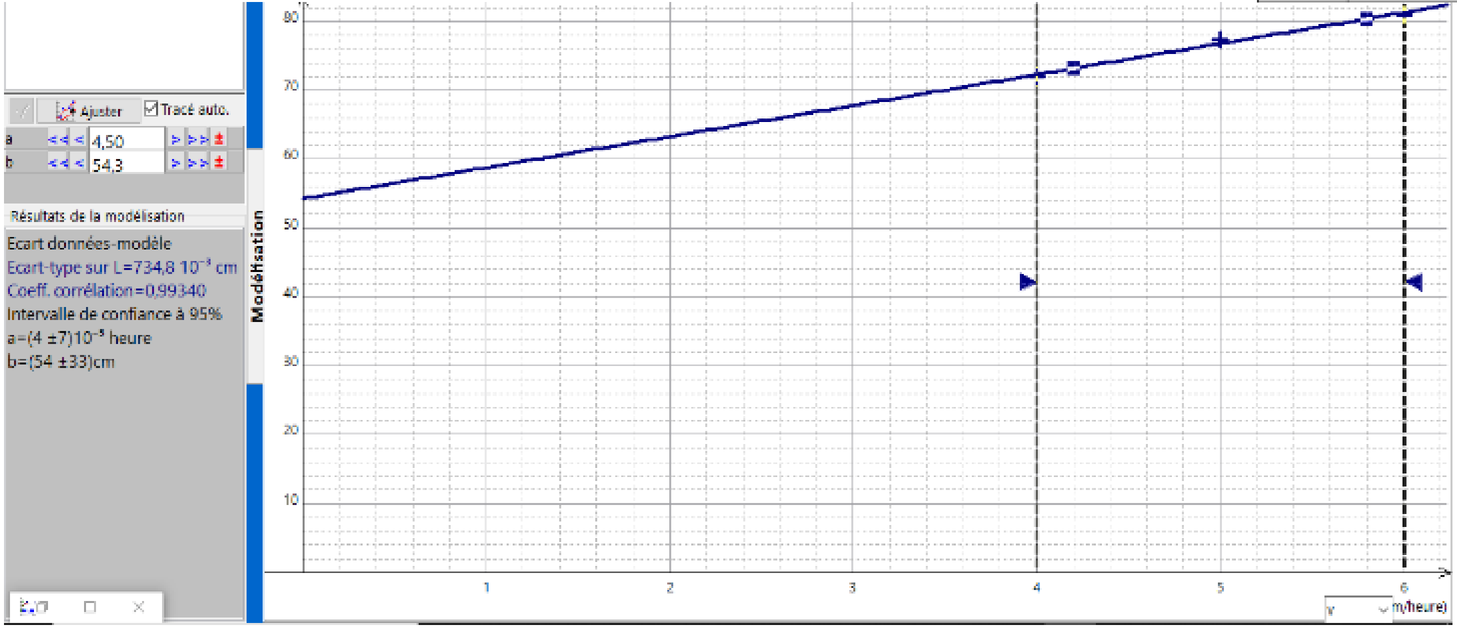
Modèle 2



taille en cm	Pas en cm à 4 km/h	Pas en cm à 5 km/h	Pas en cm à 6 km/h
150	60	64,5	67,5
155	62	66,65	69,75
160	64	68,8	72
165	66	70,95	74,25
170	68	73,1	76,5
175	70	75,25	78,75
180	72	77,4	81
185	74	79,55	83,25
190	76	81,7	85,5
195	78	83,85	87,75
200	80	86	90
205	82	88,15	92,25

L fonction linéaire de la vitesse : $L = 4.5 \times v + 54.3$

Vitesse : $v = \dfrac{\Delta S \times 54.3 \times 0.0036}{1 - 4.5 \times 0.0036 \times \Delta S}$



a 4,50

b 54,3

Résultats de la modélisation

Ecart données-modèle

Ecart-type sur L=734,8 10^{-3} cm

Coeff. corrélation=0,99340

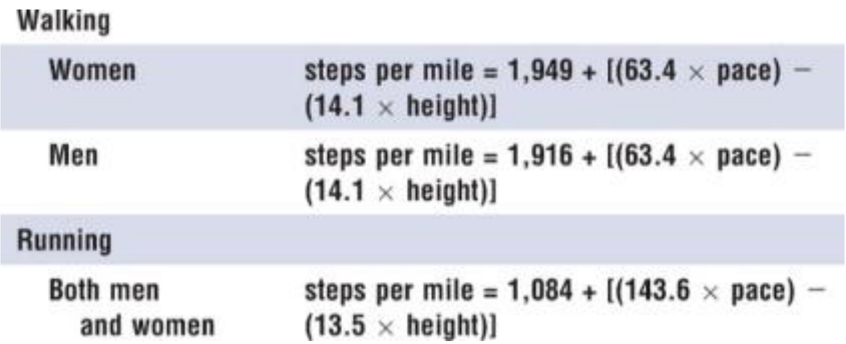
Intervalle de confiance à 95%

a=(4 ±7)10^{-5} heure

b=(54 ±33)cm

Modèle 3

L fonction de la vitesse et la taille :



Walking	
Women	steps per mile = 1,949 + [(63.4 × pace) − (14.1 × height)]
Men	steps per mile = 1,916 + [(63.4 × pace) − (14.1 × height)]
Running	
Both men and women	steps per mile = 1,084 + [(143.6 × pace) − (13.5 × height)]

Conversions : $L = \frac{1.609\times10^5}{steps\ per\ mile}$ $pace = \frac{96.56064}{v}$ $height = H\ \times 0.3937$

Vitesse :

- walking (men) $v = \frac{\Delta S\times1.609\times3600\times0.1-63.4\times96.56064}{1949-14.1\times0.3937\times H}$

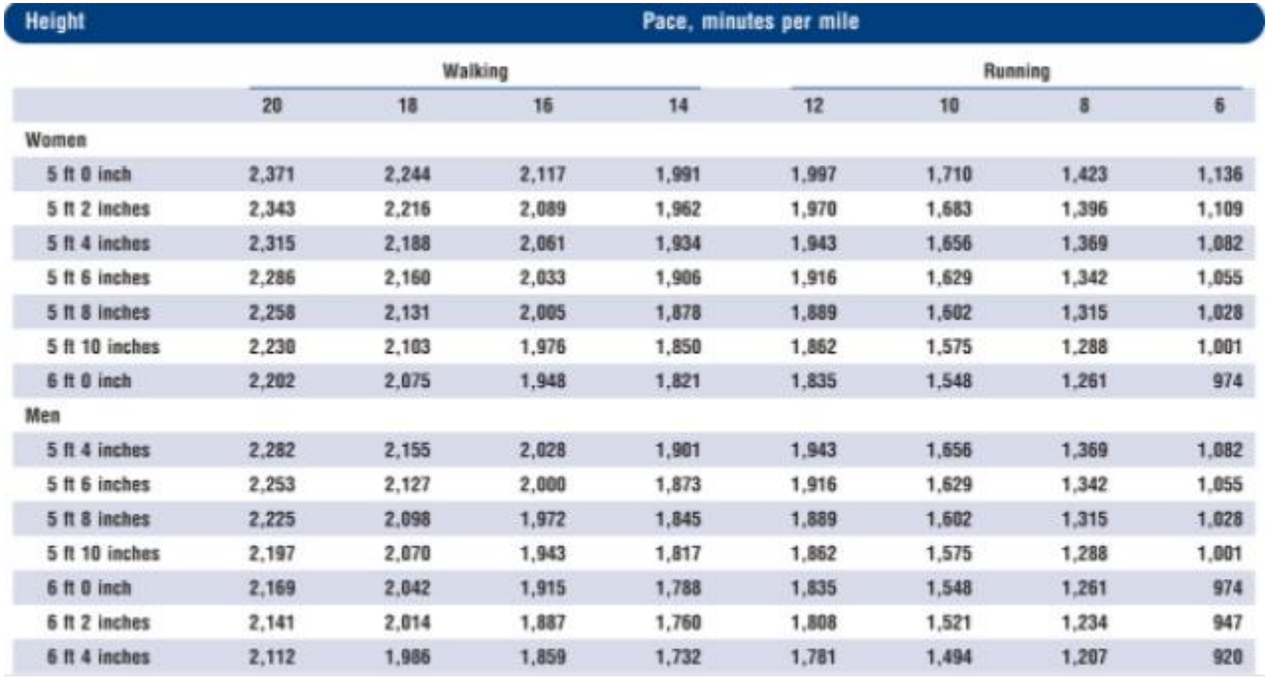
- walking (women) $v = \frac{\Delta S\times1.609\times3600\times0.1-63.4\times96.56064}{1916-14.1\times0.3937\times H}$

- running (women) $v = \frac{\Delta S\times1.609\times3600\times0.1-143.6\times96.56064}{1084-13.5\times0.3937\times H}$

Modèle 4

L fonction linéaire de la vitesse : $L = 7.2 \times v + 33.3$ (men)

$L = 7.59 \times v + 34.1$ (women)

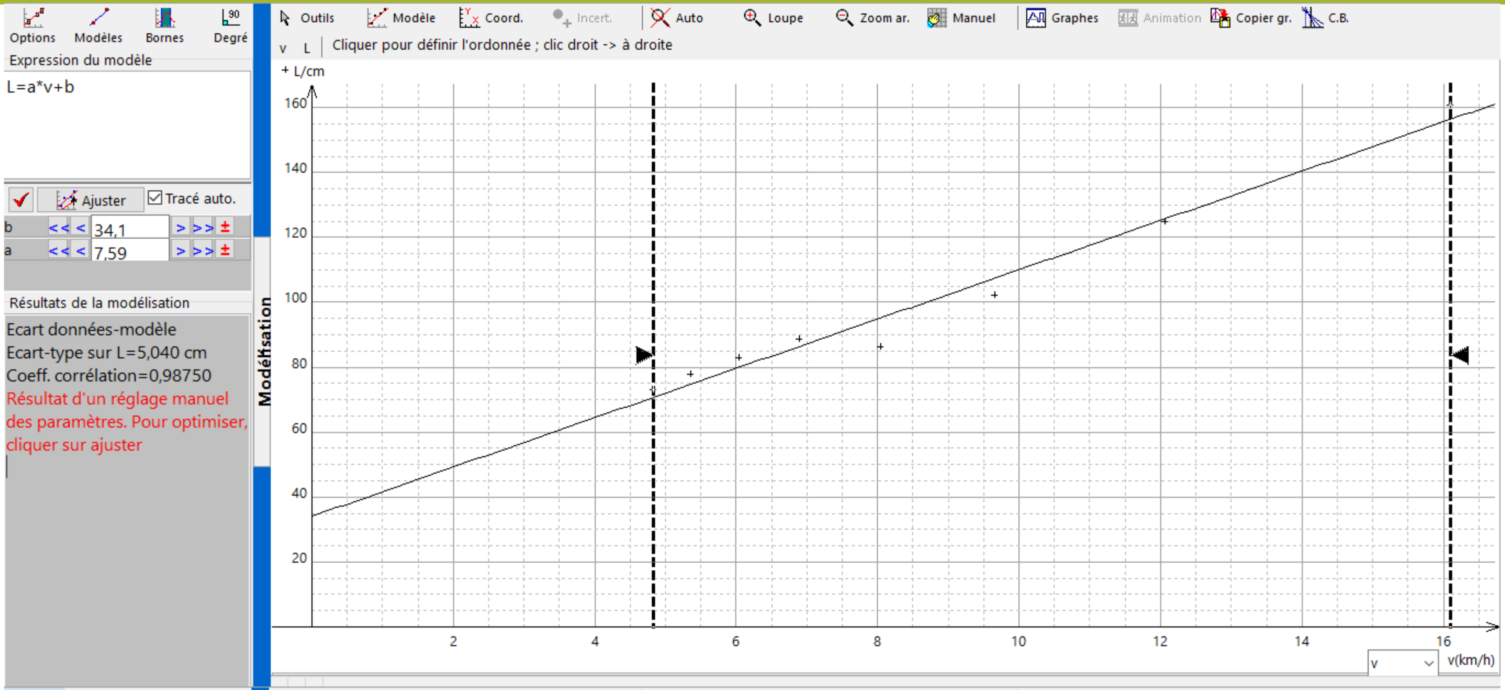


Height	Pace, minutes per mile							
	Walking				Running			
	20	18	16	14	12	10	8	6
Women								
5 ft 0 inch	2,371	2,244	2,117	1,991	1,997	1,710	1,423	1,136
5 ft 2 inches	2,343	2,216	2,089	1,962	1,970	1,683	1,396	1,109
5 ft 4 inches	2,315	2,188	2,061	1,934	1,943	1,656	1,369	1,082
5 ft 6 inches	2,286	2,160	2,033	1,906	1,916	1,629	1,342	1,055
5 ft 8 inches	2,258	2,131	2,005	1,878	1,889	1,602	1,315	1,028
5 ft 10 inches	2,230	2,103	1,976	1,850	1,862	1,575	1,288	1,001
6 ft 0 inch	2,202	2,075	1,948	1,821	1,835	1,548	1,261	974
Men								
5 ft 4 inches	2,282	2,155	2,028	1,901	1,943	1,656	1,369	1,082
5 ft 6 inches	2,253	2,127	2,000	1,873	1,916	1,629	1,342	1,055
5 ft 8 inches	2,225	2,098	1,972	1,845	1,889	1,602	1,315	1,028
5 ft 10 inches	2,197	2,070	1,943	1,817	1,862	1,575	1,288	1,001
6 ft 0 inch	2,169	2,042	1,915	1,788	1,835	1,548	1,261	974
6 ft 2 inches	2,141	2,014	1,887	1,760	1,808	1,521	1,234	947
6 ft 4 inches	2,112	1,986	1,859	1,732	1,781	1,494	1,207	920

Vitesse :

$$v = \frac{\Delta S \times 33.3 \times 0.0036}{1 - 7.2 \times 0.0036 \times \Delta S} \text{ (men)}$$

$$v = \frac{\Delta S \times 34.1 \times 0.0036}{1 - 7.59 \times 0.0036 \times \Delta S} \text{ (women)}$$



Cliquer pour définir l'ordonnée ; clic droit -> à droite

Expression du modèle

L=a*v+b

b 34,1

a 7,59

Résultats de la modélisation

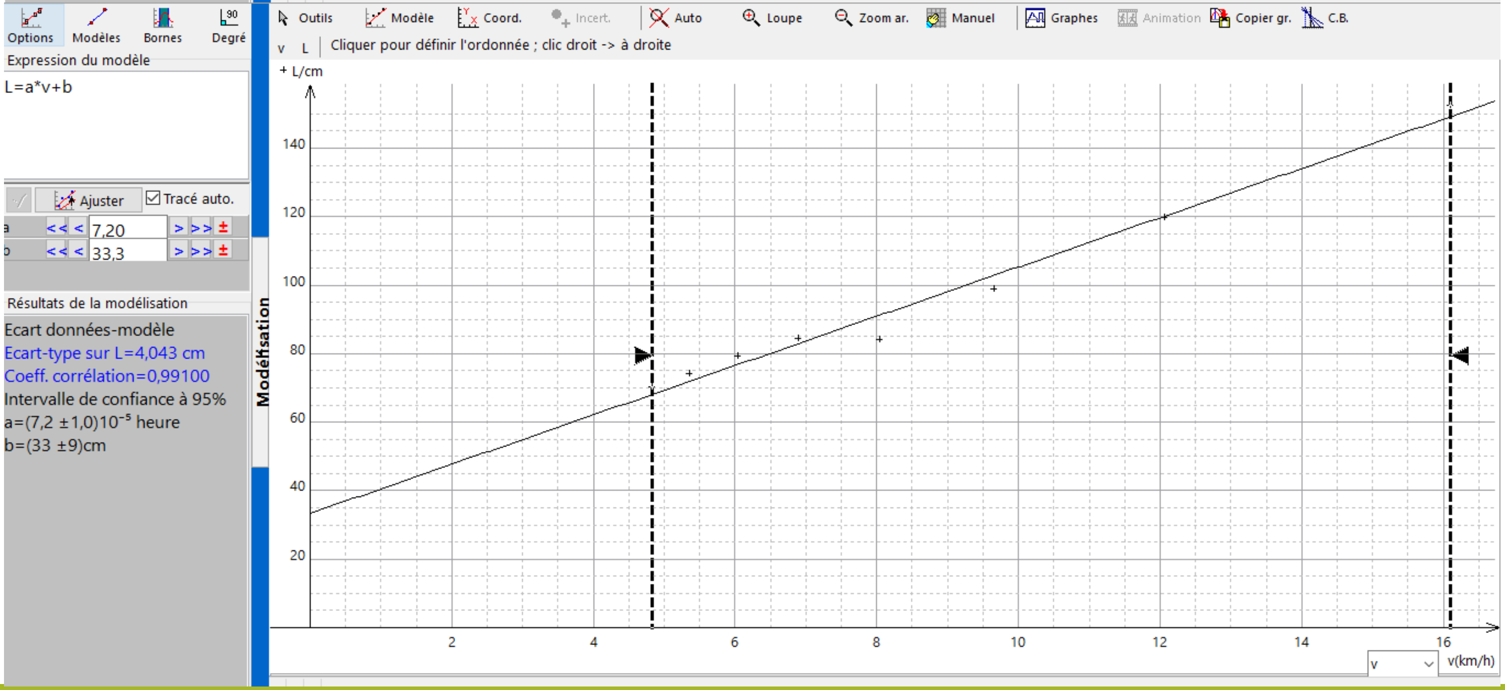
Ecart données-modèle

Ecart-type sur L=5,040 cm

Coeff. corrélation=0,98750

Résultat d'un réglage manuel des paramètres. Pour optimiser, cliquer sur ajuster

Women



Cliquer pour définir l'ordonnée ; clic droit -> à droite

Expression du modèle

L=a*v+b

a 7,20

b 33,3

Résultats de la modélisation

Ecart données-modèle

Ecart-type sur L=4,043 cm

Coeff. corrélation=0,99100

Intervalle de confiance à 95%

a=(7,2 ±1,0)10^{-5} heure

b=(33 ±9)cm

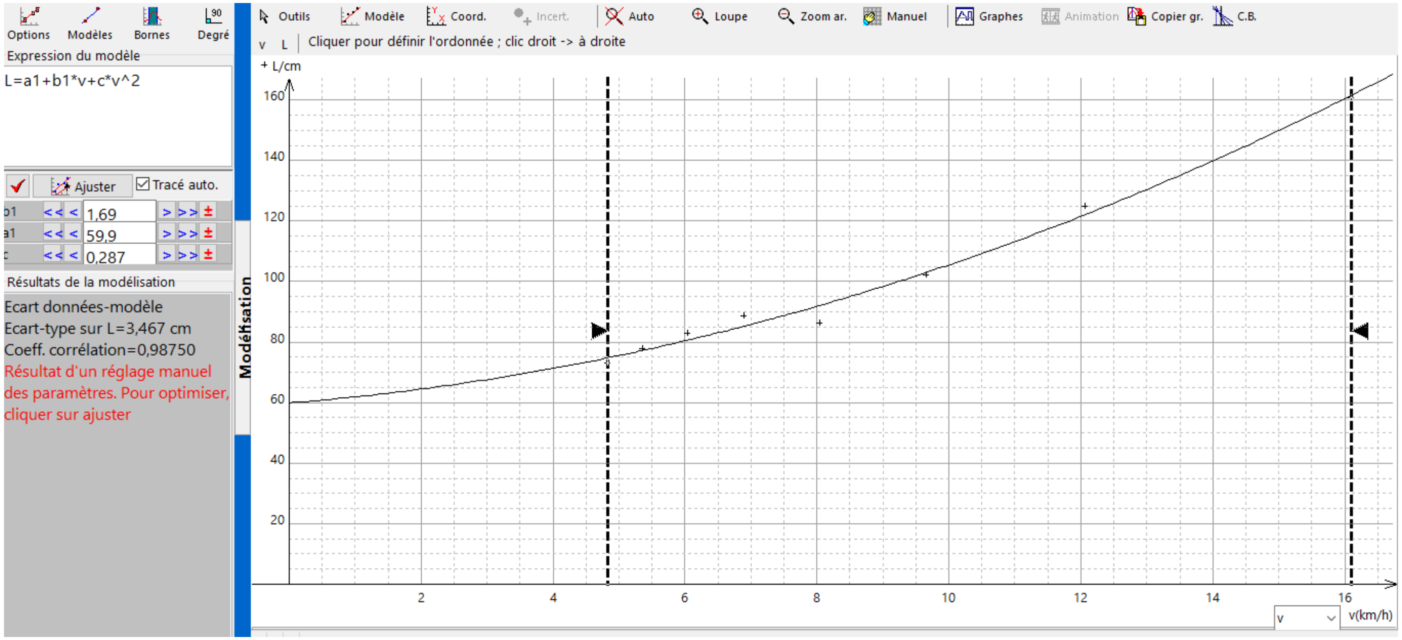
Men

Modèle 5

L fonction parabolique de la vitesse :

$$L = 0.226 \ \times \ v^2 \ + \ 2.55 \ \times v + 53.7 \text{ (men)}$$

$$L = 0.287 \ \times \ v^2 \ + 1.69 \ \times v + 59.9 \text{ (women)}$$



Cliquer pour définir l'ordonnée ; clic droit -> à droite

Expression du modèle

L=a1+b1*v+c*v^2

✓ Ajuster ☑ Tracé auto.

b1	1,69
a1	59,9
c	0,287

Résultats de la modélisation

Ecart données-modèle
Ecart-type sur L=3,467 cm
Coeff. corrélation=0,98750
Résultat d'un réglage manuel des paramètres. Pour optimiser, cliquer sur ajuster

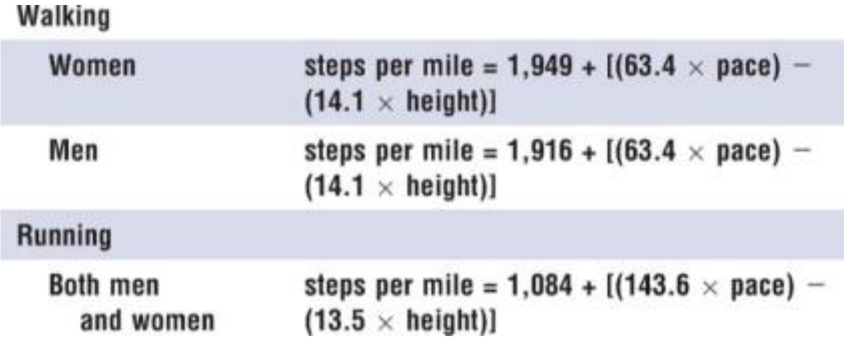
Women

$$v = \frac{1 - 0.0036 \times 2.55 \times \Delta S - \sqrt{(0.0036 \times 2.55 \times \Delta S - 1)^2 - 4(0.0036 \times \Delta S)^2 \times 59.9 \times 0.226}}{2 \times 0.0036 \times 0.226 \times \Delta S}$$

(women)

Modèle 6

L fonction de la vitesse et la taille :



Walking	
Women	steps per mile = 1,949 + [(63.4 × pace) − (14.1 × height)]
Men	steps per mile = 1,916 + [(63.4 × pace) − (14.1 × height)]
Running	
Both men and women	steps per mile = 1,084 + [(143.6 × pace) − (13.5 × height)]

Vitesse : - walking (men)

$$v = \frac{\Delta S \times 1.609 \times 3600 \times 0.1 - 63.4 \times 96.56064}{1949 - 14.1 \times 0.3937 \times H}$$

- walking (women)

$$v = \frac{\Delta S \times 1.609 \times 3600 \times 0.1 - 63.4 \times 96.56064}{1916 - 14.1 \times 0.3937 \times H}$$

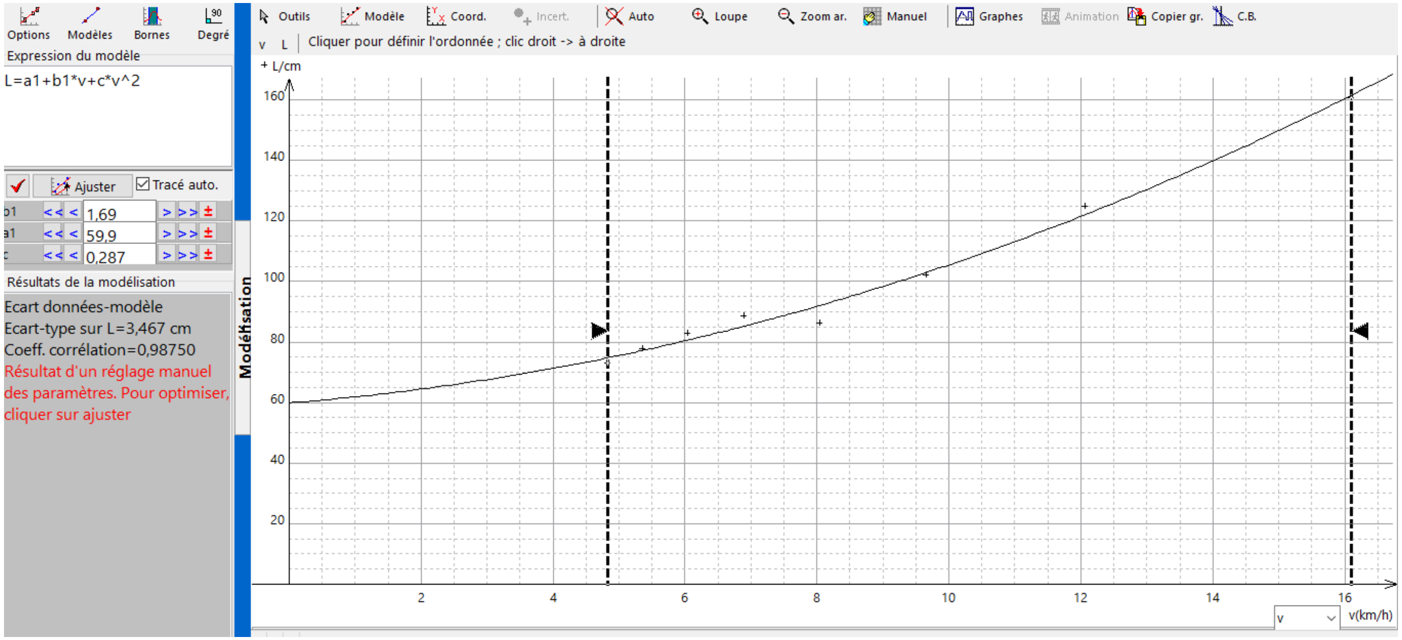
On ne prend que « walking » (l’autre créé des valeurs trop grandes) et pour $\Delta S \ < 11$ on dit que v = 0 (éviter valeurs négatives)

Modèle 7

L fonction parabolique de la vitesse :

$$L = 0.226 \times v^2 + 2.55 \times v + 53.7 \text{ (men)}$$

$$L = 0.287 \times v^2 + 1.69 \times v + 59.9 \text{ (women)}$$



Cliquer pour définir l'ordonnée ; clic droit -> à droite

Expression du modèle

L=a1+b1*v+c*v^2

✓ Ajuster ☑ Tracé auto.

b1	1,69
a1	59,9
c	0,287

Résultats de la modélisation

Ecart données-modèle
Ecart-type sur L=3,467 cm
Coeff. corrélation=0,98750
Résultat d'un réglage manuel des paramètres. Pour optimiser, cliquer sur ajuster

Women

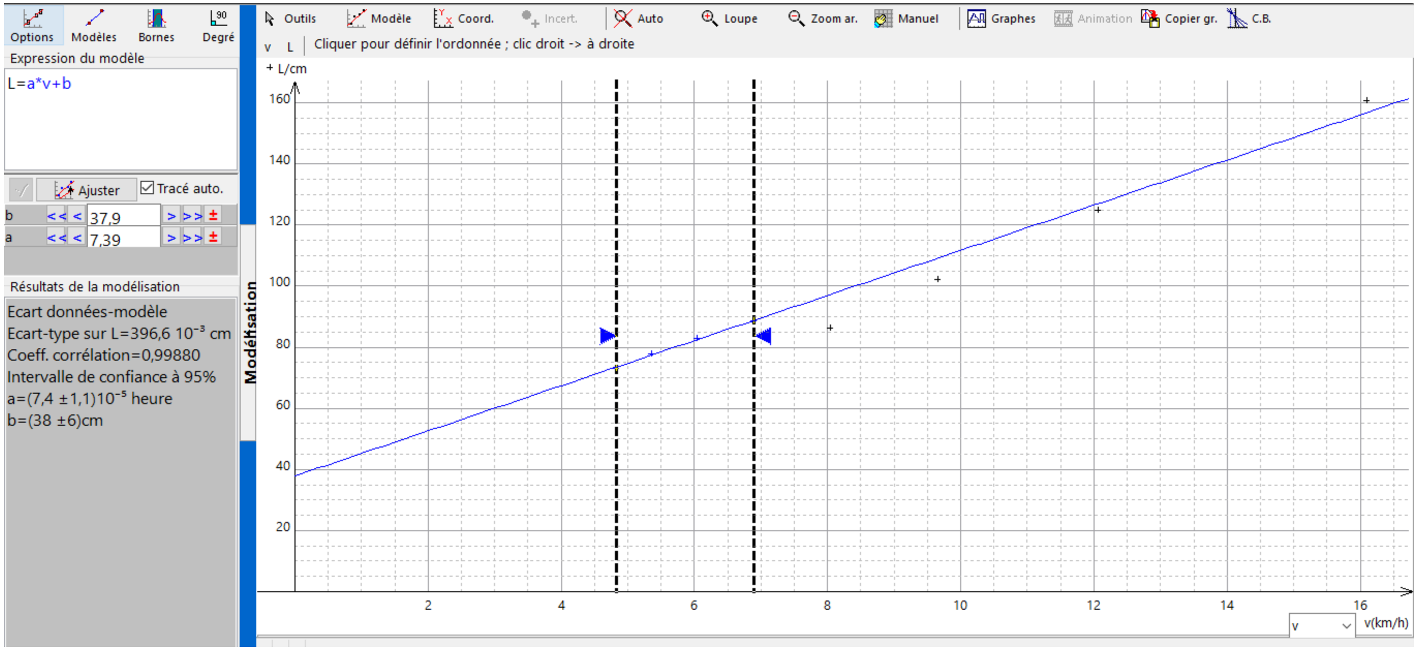
$$v = \frac{1 - 0.0036 \times 2.55 \times \Delta S - \sqrt{(0.0036 \times 2.55 \times \Delta S - 1)^2 - 4(0.0036 \times \Delta S)^2 \times 59.9 \times 0.226}}{2 \times 0.0036 \times 0.226 \times \Delta S}$$

Si $\Delta S > 29$ on pose la racine nulle (éviter les racines de négatif)

Modèle 8

L fonction linéaire de la vitesse : $L = 6.85 \times v + 37.4$ (men)

$L = 7.39 \times v + 37.9$ (women)



Expression du modèle

L=a*v+b

☑ Tracé auto.

b 37,9

a 7,39

Résultats de la modélisation

Ecart données-modèle

Ecart-type sur L=396,6 10^{-3} cm

Coeff. corrélation=0,99880

Intervalle de confiance à 95%

a=(7,4 ±1,1)10^{-5} heure

b=(38 ±6)cm

(women)

On ne prend que les valeurs pour la marche

Vitesse :

$$v = \frac{\Delta S \times 37.4 \times 0.0036}{1 - 6.85 \times 0.0036 \times \Delta S} \text{ (men)}$$

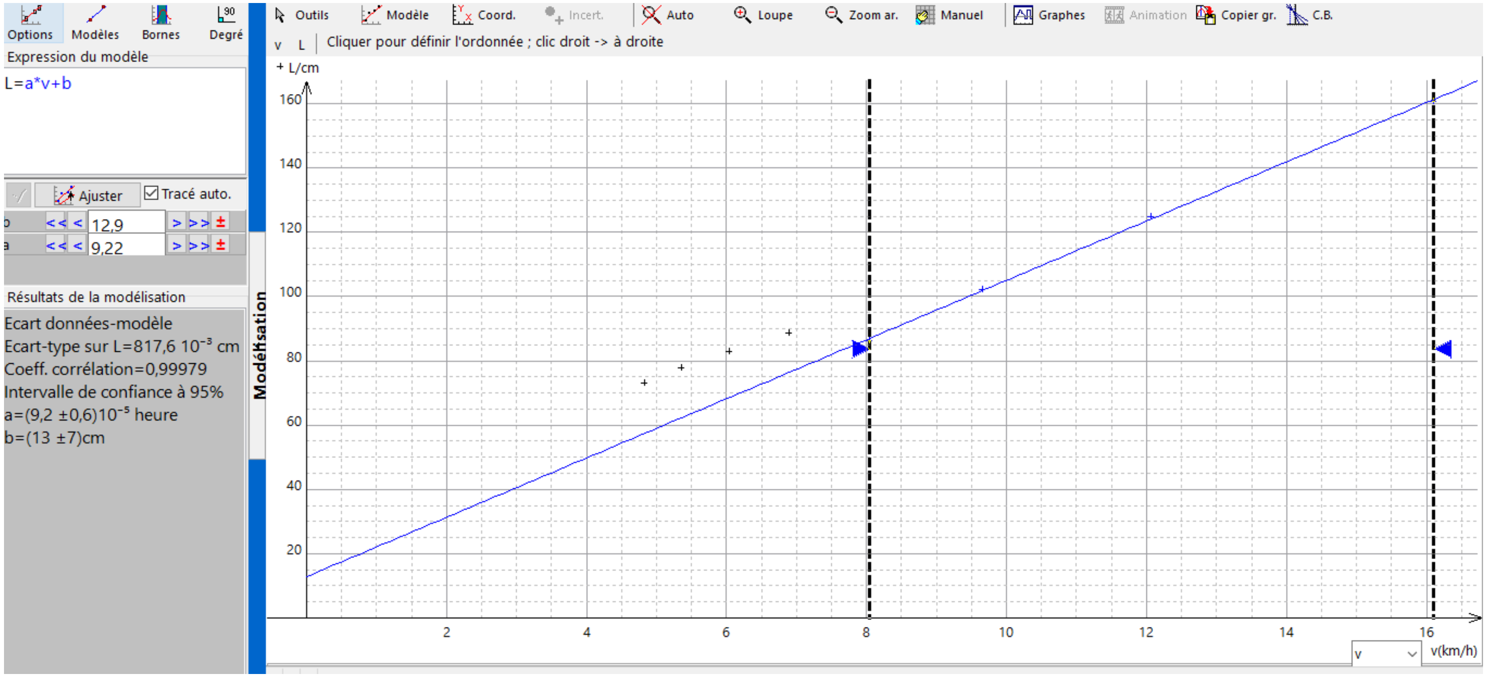
$$v = \frac{\Delta S \times 37.9 \times 0.0036}{1 - 7.39 \times 0.0036 \times \Delta S} \text{ (women)}$$

Modèle 9

L fonction linéaire de la vitesse :

$$L = 8.49 \times v + 16.4 \text{ (men)}$$

$$L = 9.22 \times v + 12.9 \text{ (women)}$$



Expression du modèle

L=a*v+b

b 12,9

a 9,22

Résultats de la modélisation

Ecart données-modèle
Ecart-type sur L=817,6 10^{-3} cm
Coeff. corrélation=0,99979
Intervalle de confiance à 95%
a=(9,2 ±0,6)10^{-5} heure
b=(13 ±7)cm

(women)

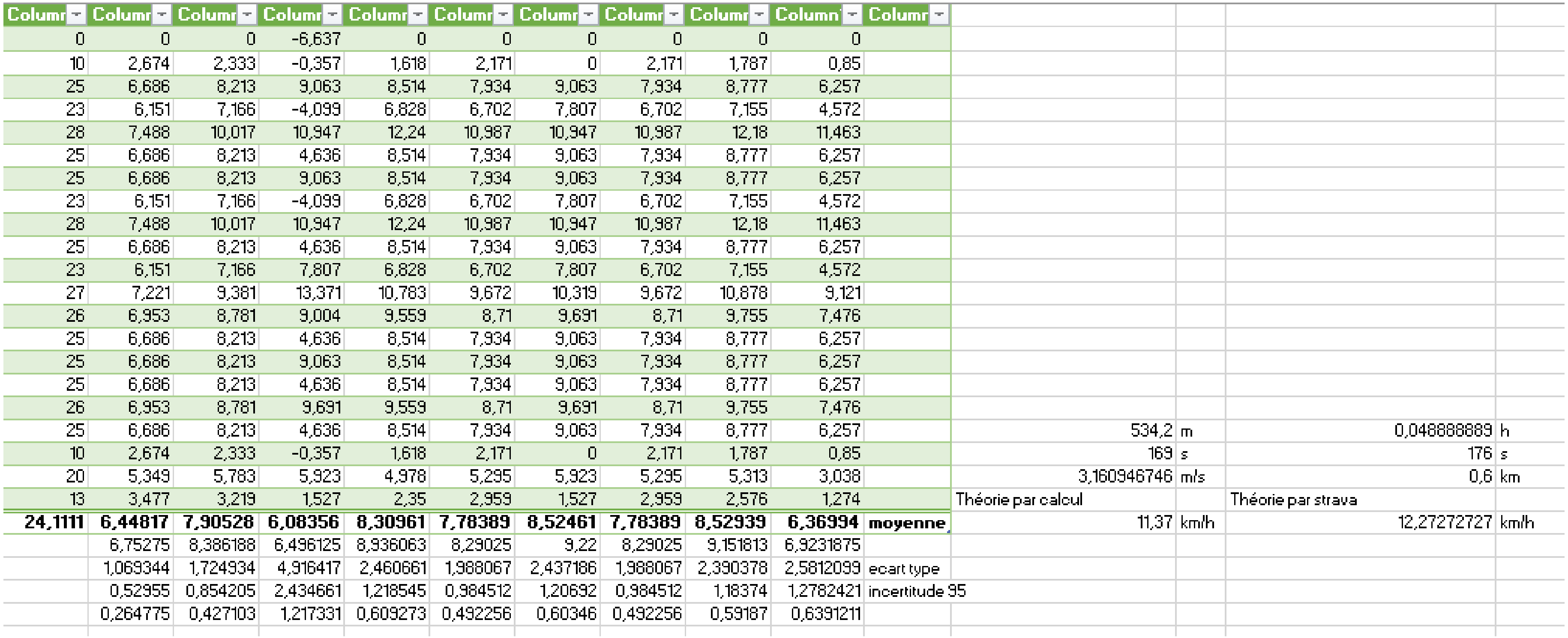
On ne prend que les valeurs pour la course

Vitesse :

$$v = \frac{\Delta S \times 16.4 \times 0.0036}{1 - 8.49 \times 0.0036 \times \Delta S} \text{ (men)}$$

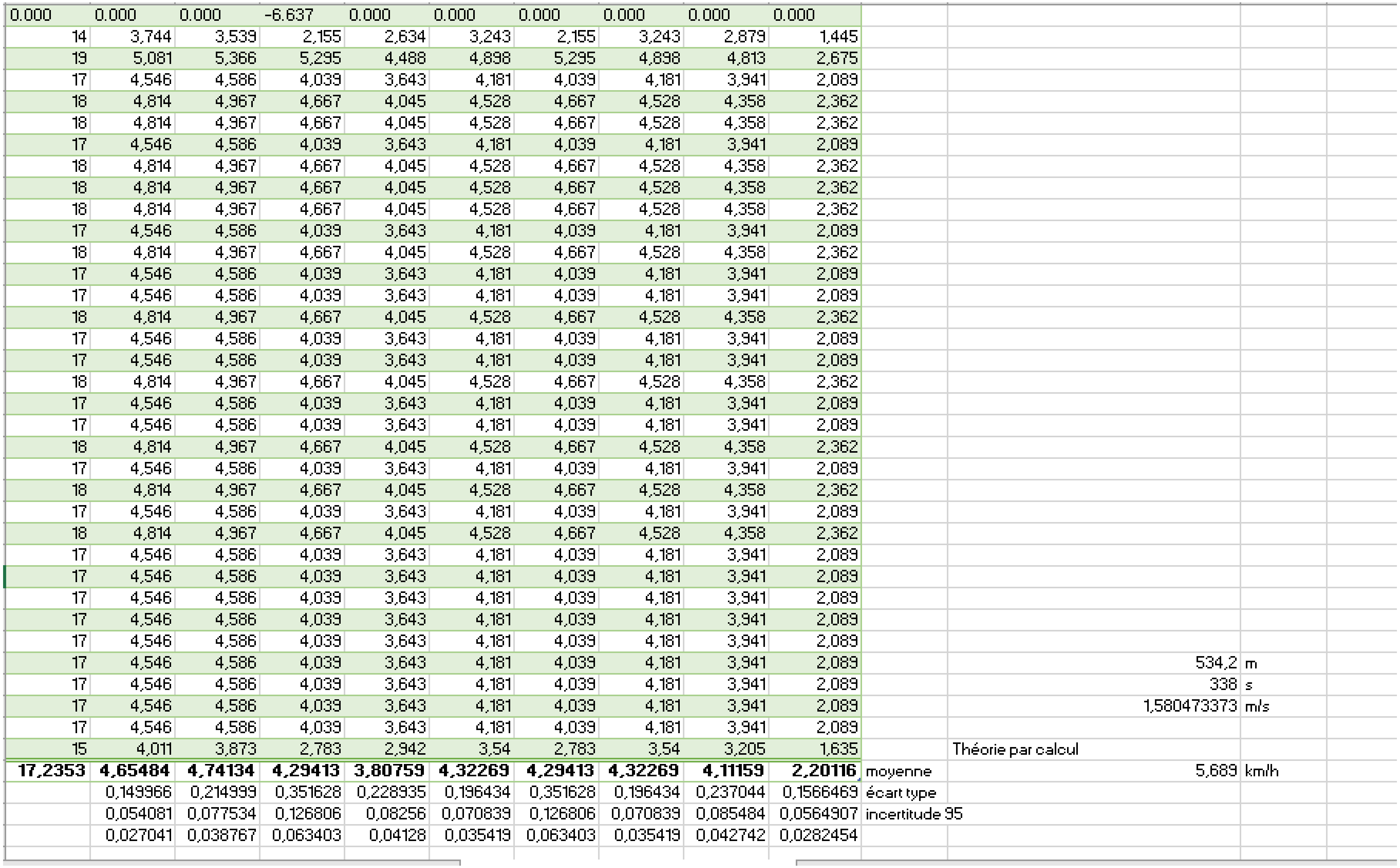
$$v = \frac{\Delta S \times 12.9 \times 0.0036}{1 - 9.22 \times 0.0036 \times \Delta S} \text{ (women)}$$

Course 2 tours de terrain

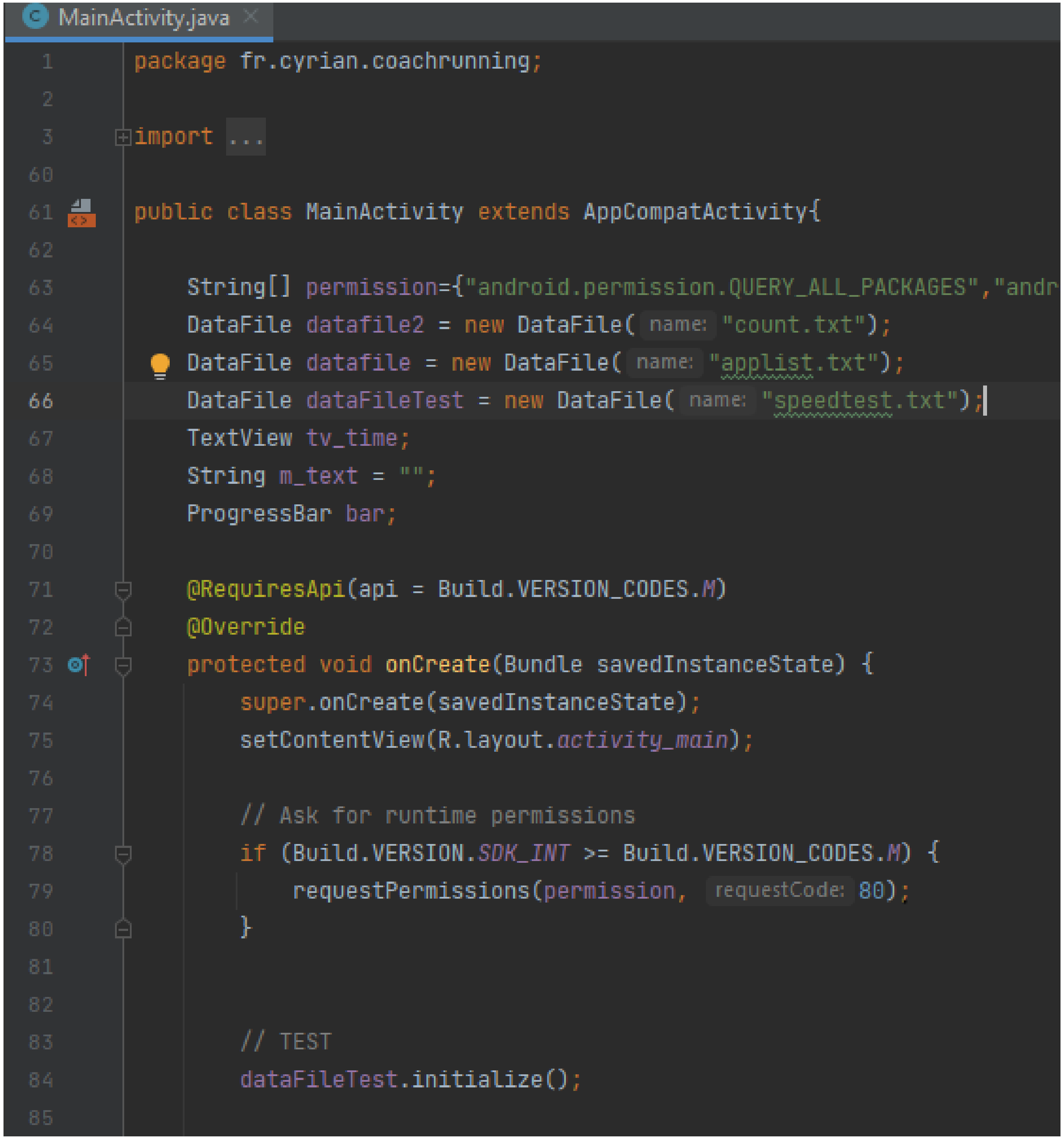


Columr	Columr	Columr	Columr	Columr	Columr	Columr	Columr	Columr	Column	Columr				
0	0	0	-6,637	0	0	0	0	0	0					
10	2,674	2,333	-0,357	1,618	2,171	0	2,171	1,787	0,85					
25	6,686	8,213	9,063	8,514	7,934	9,063	7,934	8,777	6,257					
23	6,151	7,166	-4,099	6,828	6,702	7,807	6,702	7,155	4,572					
28	7,488	10,017	10,947	12,24	10,987	10,947	10,987	12,18	11,463					
25	6,686	8,213	4,636	8,514	7,934	9,063	7,934	8,777	6,257					
25	6,686	8,213	9,063	8,514	7,934	9,063	7,934	8,777	6,257					
23	6,151	7,166	-4,099	6,828	6,702	7,807	6,702	7,155	4,572					
28	7,488	10,017	10,947	12,24	10,987	10,947	10,987	12,18	11,463					
25	6,686	8,213	4,636	8,514	7,934	9,063	7,934	8,777	6,257					
23	6,151	7,166	7,807	6,828	6,702	7,807	6,702	7,155	4,572					
27	7,221	9,381	13,371	10,783	9,672	10,319	9,672	10,878	9,121					
26	6,953	8,781	9,004	9,559	8,71	9,691	8,71	9,755	7,476					
25	6,686	8,213	4,636	8,514	7,934	9,063	7,934	8,777	6,257					
25	6,686	8,213	9,063	8,514	7,934	9,063	7,934	8,777	6,257					
25	6,686	8,213	4,636	8,514	7,934	9,063	7,934	8,777	6,257					
26	6,953	8,781	9,691	9,559	8,71	9,691	8,71	9,755	7,476					
25	6,686	8,213	4,636	8,514	7,934	9,063	7,934	8,777	6,257		534,2	m	0,048888889	h
10	2,674	2,333	-0,357	1,618	2,171	0	2,171	1,787	0,85		169	s	176	s
20	5,349	5,783	5,923	4,978	5,295	5,923	5,295	5,313	3,038		3,160946746	m/s	0,6	km
13	3,477	3,219	1,527	2,35	2,959	1,527	2,959	2,576	1,274		Théorie par calcul		Théorie par strava	
24,1111	**6,44817**	**7,90528**	**6,08356**	**8,30961**	**7,78389**	**8,52461**	**7,78389**	**8,52939**	**6,36994**	**moyenne**	11,37	km/h	12,27272727	km/h
	6,75275	8,386188	6,496125	8,936063	8,29025	9,22	8,29025	9,151813	6,9231875					
	1,069344	1,724934	4,916417	2,460661	1,988067	2,437186	1,988067	2,390378	2,5812099	ecart type				
	0,52955	0,854205	2,434661	1,218545	0,984512	1,20692	0,984512	1,18374	1,2782421	incertitude 95				
	0,264775	0,427103	1,217331	0,609273	0,492256	0,60346	0,492256	0,59187	0,6391211					

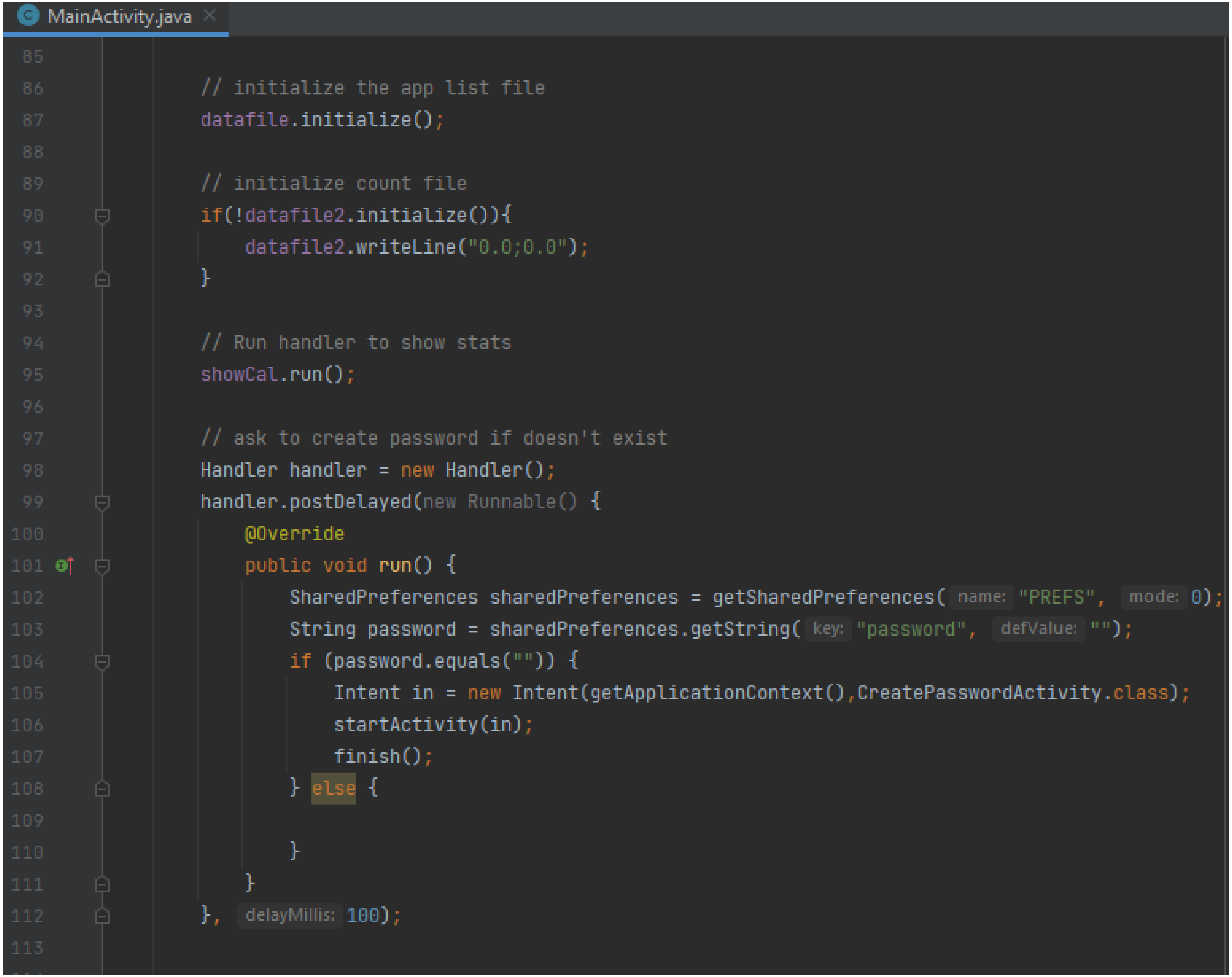
Marche 2 tours de terrain



0.000	0.000	0.000	-6.637	0.000	0.000	0.000	0.000	0.000	0.000			
14	3,744	3,539	2,155	2,634	3,243	2,155	3,243	2,879	1,445			
19	5,081	5,366	5,295	4,488	4,898	5,295	4,898	4,813	2,675			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
18	4,814	4,967	4,667	4,045	4,528	4,667	4,528	4,358	2,362			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089		534,2	m
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089		338	s
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089		1,580473373	m/s
17	4,546	4,586	4,039	3,643	4,181	4,039	4,181	3,941	2,089			
15	4,011	3,873	2,783	2,942	3,54	2,783	3,54	3,205	1,635		Théorie par calcul	
17,2353	**4,65484**	**4,74134**	**4,29413**	**3,80759**	**4,32269**	**4,29413**	**4,32269**	**4,11159**	**2,20116**	moyenne	5,689	km/h
	0,149966	0,214999	0,351628	0,228935	0,196434	0,351628	0,196434	0,237044	0,1566469	écart type		
	0,054081	0,077534	0,126806	0,08256	0,070839	0,126806	0,070839	0,085484	0,0564907	incertitude 95		
	0,027041	0,038767	0,063403	0,04128	0,035419	0,063403	0,035419	0,042742	0,0282454			

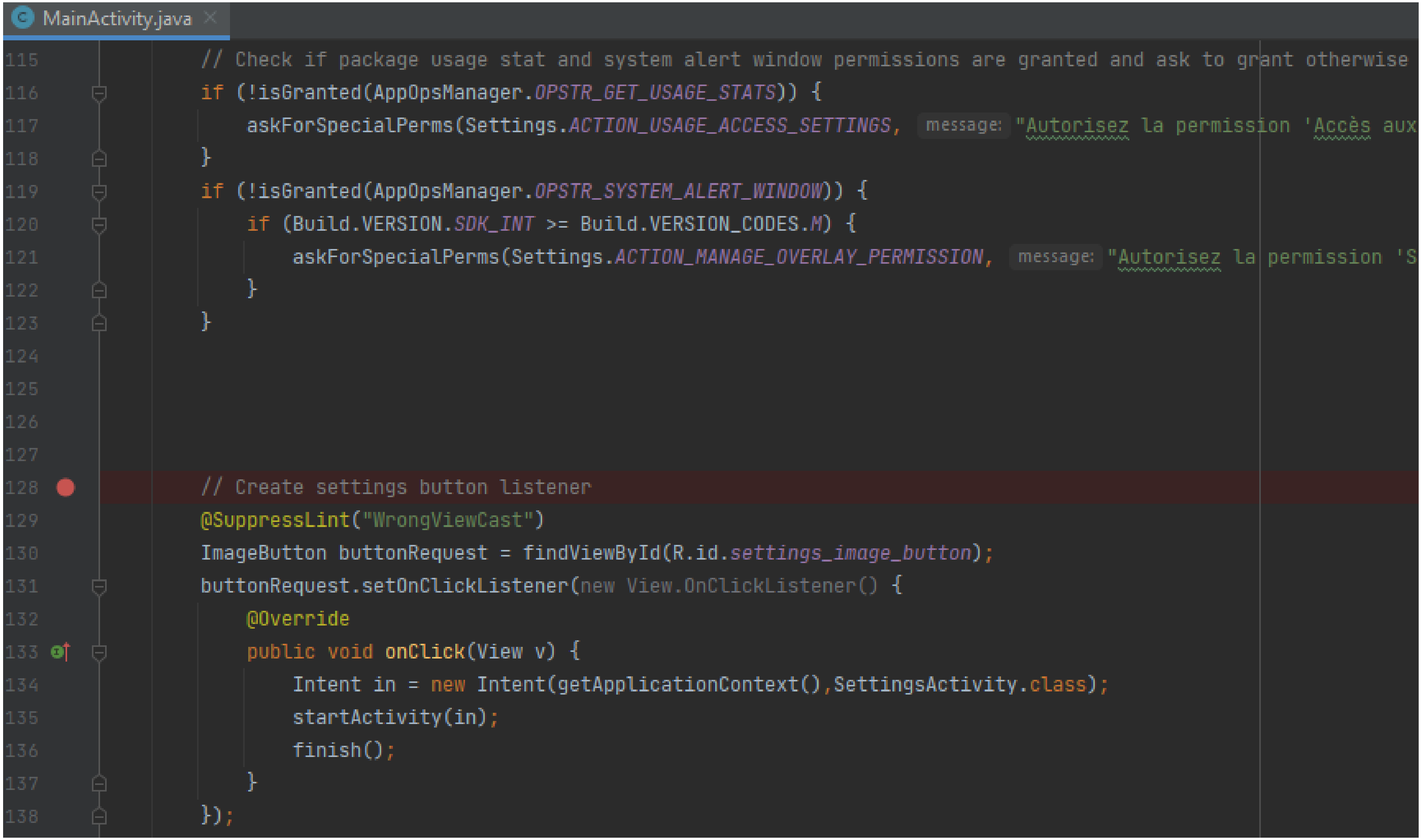


```
package fr.cyrian.coachrunning;

import ...

public class MainActivity extends AppCompatActivity{

    String[] permission={"android.permission.QUERY_ALL_PACKAGES","andr
    DataFile datafile2 = new DataFile( name: "count.txt");
    DataFile datafile = new DataFile( name: "applist.txt");
    DataFile dataFileTest = new DataFile( name: "speedtest.txt");
    TextView tv_time;
    String m_text = "";
    ProgressBar bar;

    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Ask for runtime permissions
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(permission, requestCode: 80);
        }


        // TEST
        dataFileTest.initialize();

```

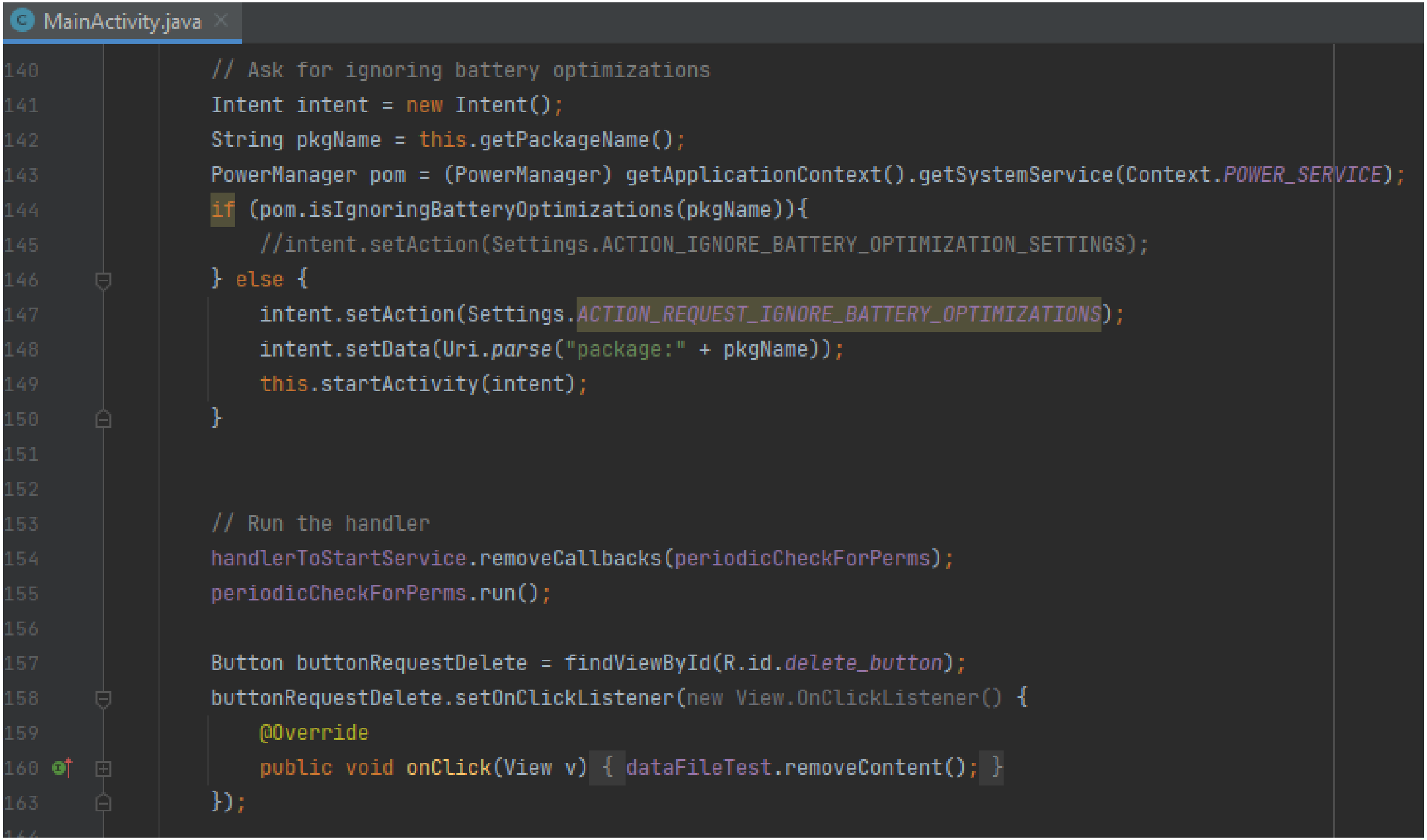


MainActivity.java

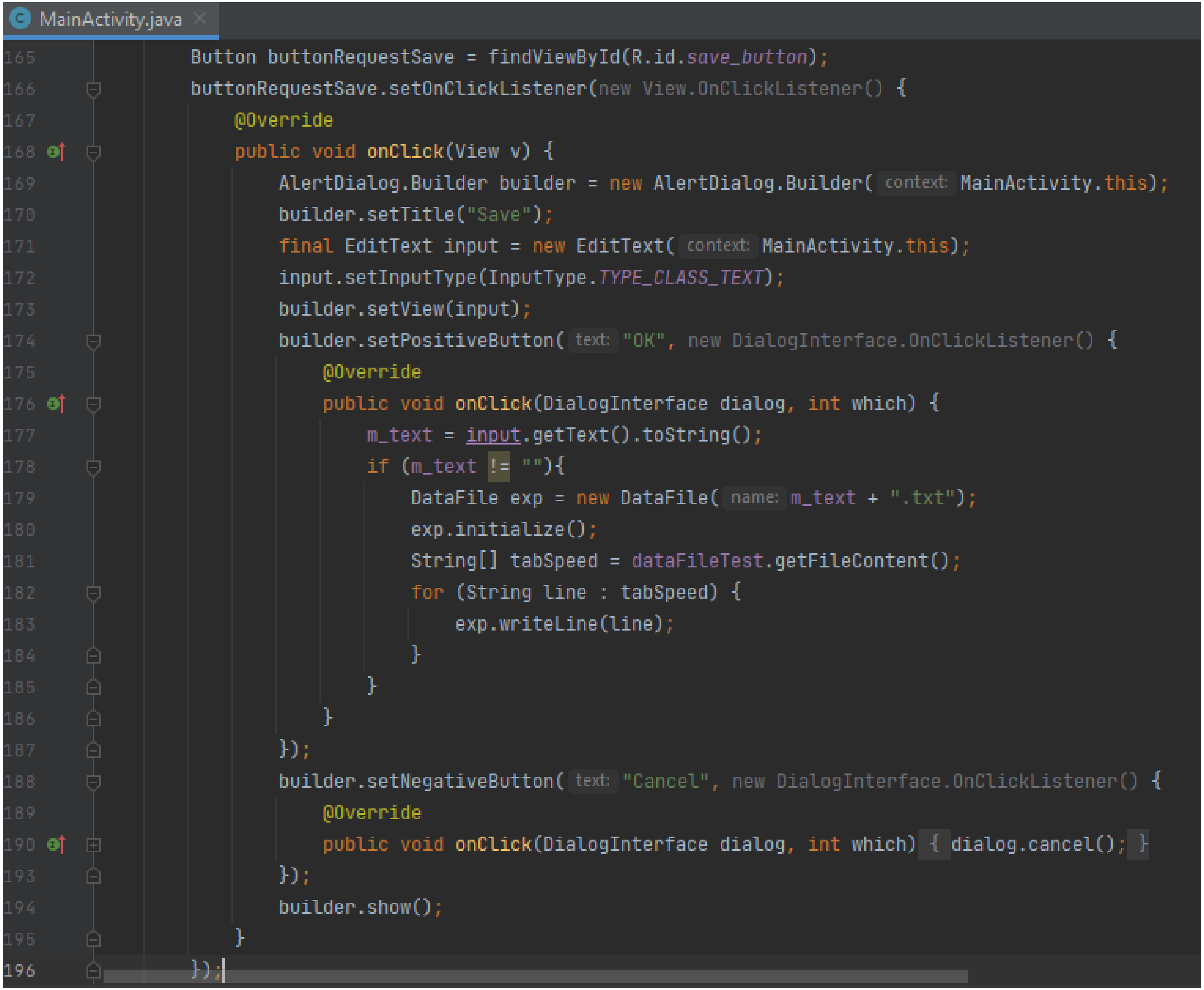
```java

        // initialize the app list file
        datafile.initialize();

        // initialize count file
        if(!datafile2.initialize()){
            datafile2.writeLine("0.0;0.0");
        }

        // Run handler to show stats
        showCal.run();

        // ask to create password if doesn't exist
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                SharedPreferences sharedPreferences = getSharedPreferences( name: "PREFS",  mode: 0);
                String password = sharedPreferences.getString( key: "password",  defValue: "");
                if (password.equals("")) {
                    Intent in = new Intent(getApplicationContext(),CreatePasswordActivity.class);
                    startActivity(in);
                    finish();
                } else {

                }
            }
        },  delayMillis: 100);

```



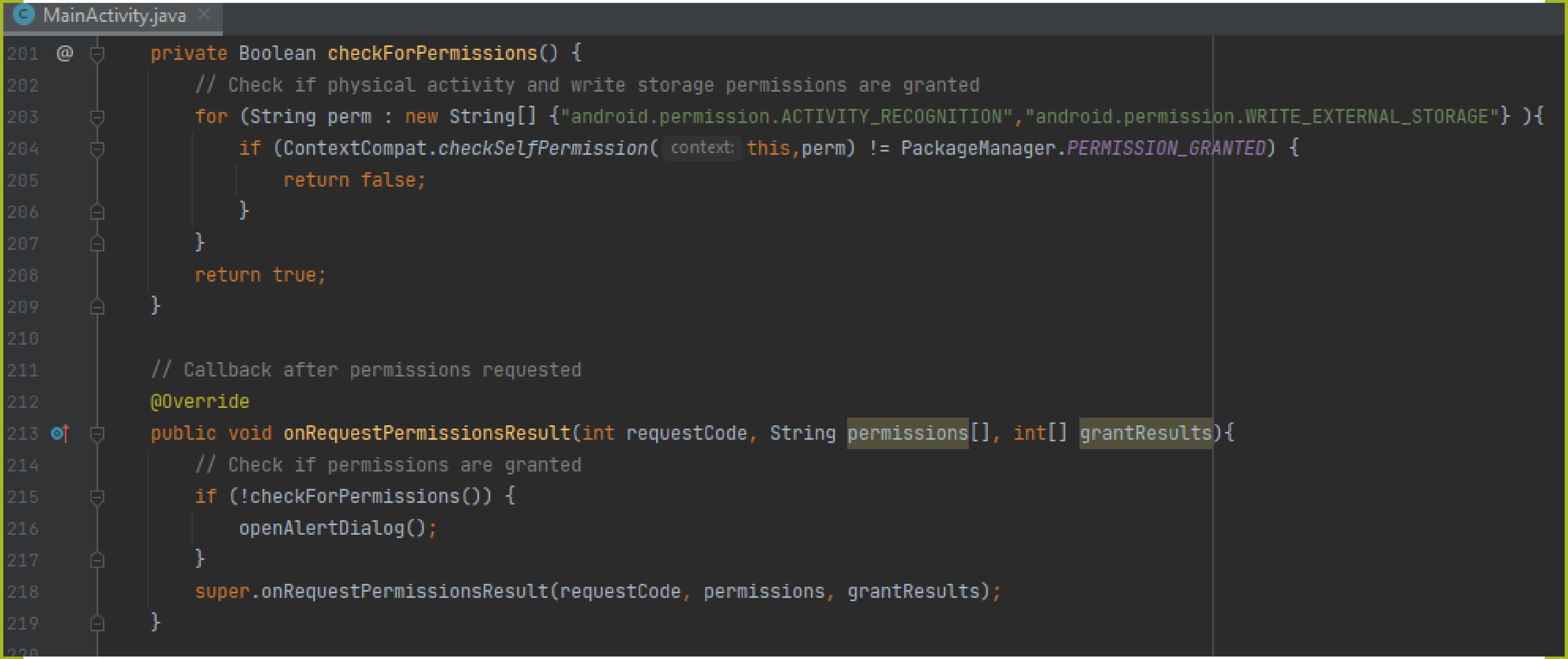
```java
        // Check if package usage stat and system alert window permissions are granted and ask to grant otherwise
        if (!isGranted(AppOpsManager.OPSTR_GET_USAGE_STATS)) {
            askForSpecialPerms(Settings.ACTION_USAGE_ACCESS_SETTINGS,  message: "Autorisez la permission 'Accès aux
        }
        if (!isGranted(AppOpsManager.OPSTR_SYSTEM_ALERT_WINDOW)) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                askForSpecialPerms(Settings.ACTION_MANAGE_OVERLAY_PERMISSION,  message: "Autorisez la permission 'S
            }
        }




        // Create settings button listener
        @SuppressLint("WrongViewCast")
        ImageButton buttonRequest = findViewById(R.id.settings_image_button);
        buttonRequest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent(getApplicationContext(),SettingsActivity.class);
                startActivity(in);
                finish();
            }
        });
```



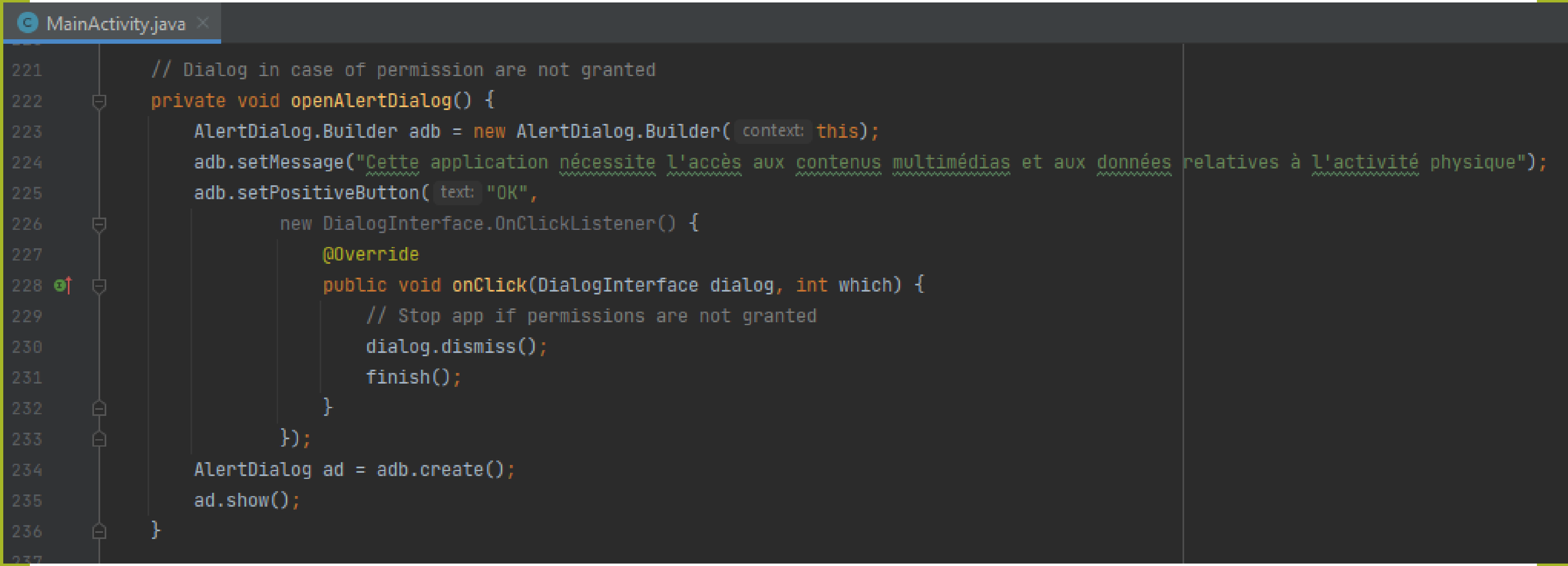
```java
        // Ask for ignoring battery optimizations
        Intent intent = new Intent();
        String pkgName = this.getPackageName();
        PowerManager pom = (PowerManager) getApplicationContext().getSystemService(Context.POWER_SERVICE);
        if (pom.isIgnoringBatteryOptimizations(pkgName)){
            //intent.setAction(Settings.ACTION_IGNORE_BATTERY_OPTIMIZATION_SETTINGS);
        } else {
            intent.setAction(Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS);
            intent.setData(Uri.parse("package:" + pkgName));
            this.startActivity(intent);
        }


        // Run the handler
        handlerToStartService.removeCallbacks(periodicCheckForPerms);
        periodicCheckForPerms.run();

        Button buttonRequestDelete = findViewById(R.id.delete_button);
        buttonRequestDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { dataFileTest.removeContent(); }
        });
```



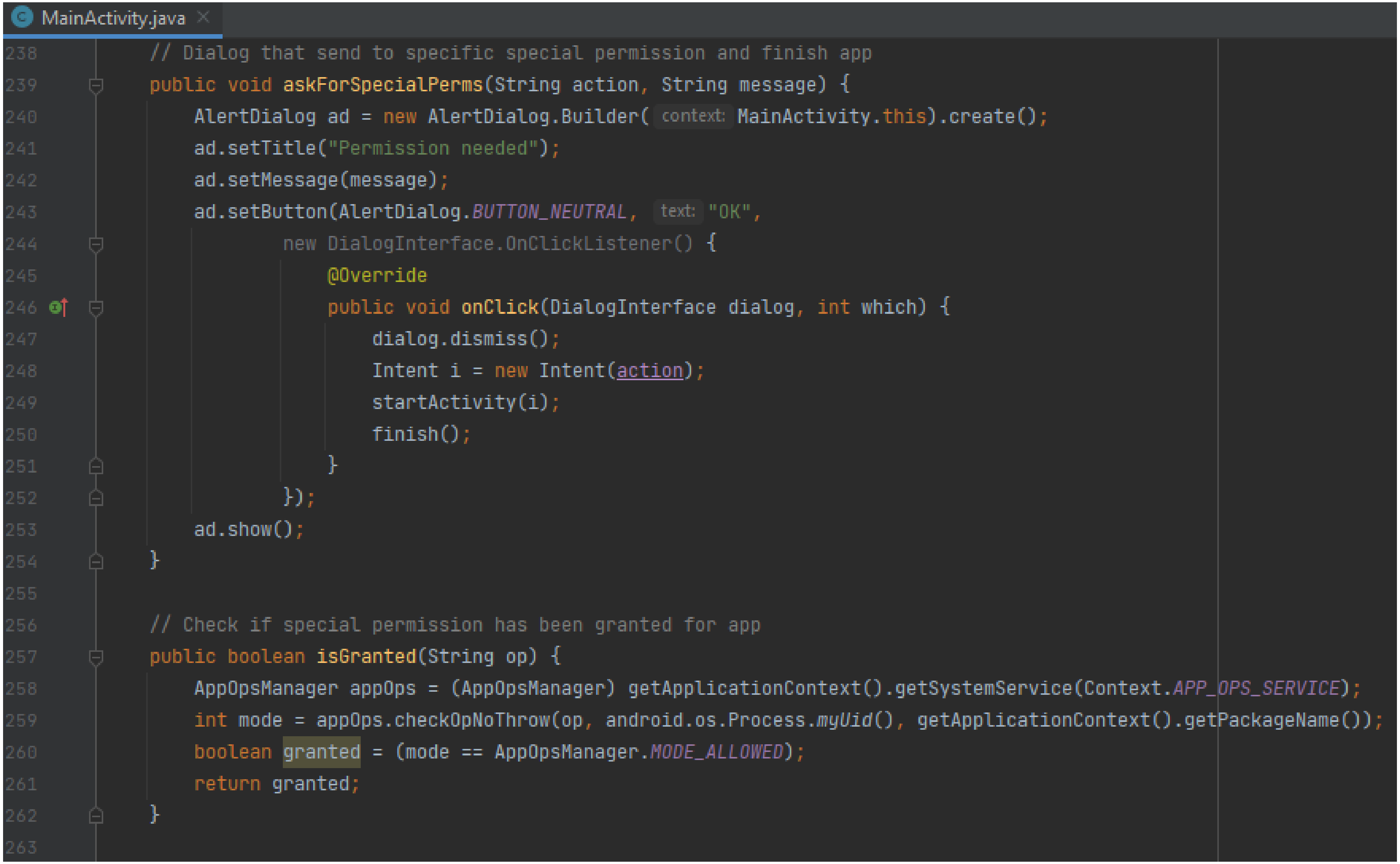
```java
        Button buttonRequestSave = findViewById(R.id.save_button);
        buttonRequestSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
                builder.setTitle("Save");
                final EditText input = new EditText( context: MainActivity.this);
                input.setInputType(InputType.TYPE_CLASS_TEXT);
                builder.setView(input);
                builder.setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        m_text = input.getText().toString();
                        if (m_text != ""){
                            DataFile exp = new DataFile( name: m_text + ".txt");
                            exp.initialize();
                            String[] tabSpeed = dataFileTest.getFileContent();
                            for (String line : tabSpeed) {
                                exp.writeLine(line);
                            }
                        }
                    }
                });
                builder.setNegativeButton( text: "Cancel", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) { dialog.cancel(); }
                });
                builder.show();
            }
        });
```



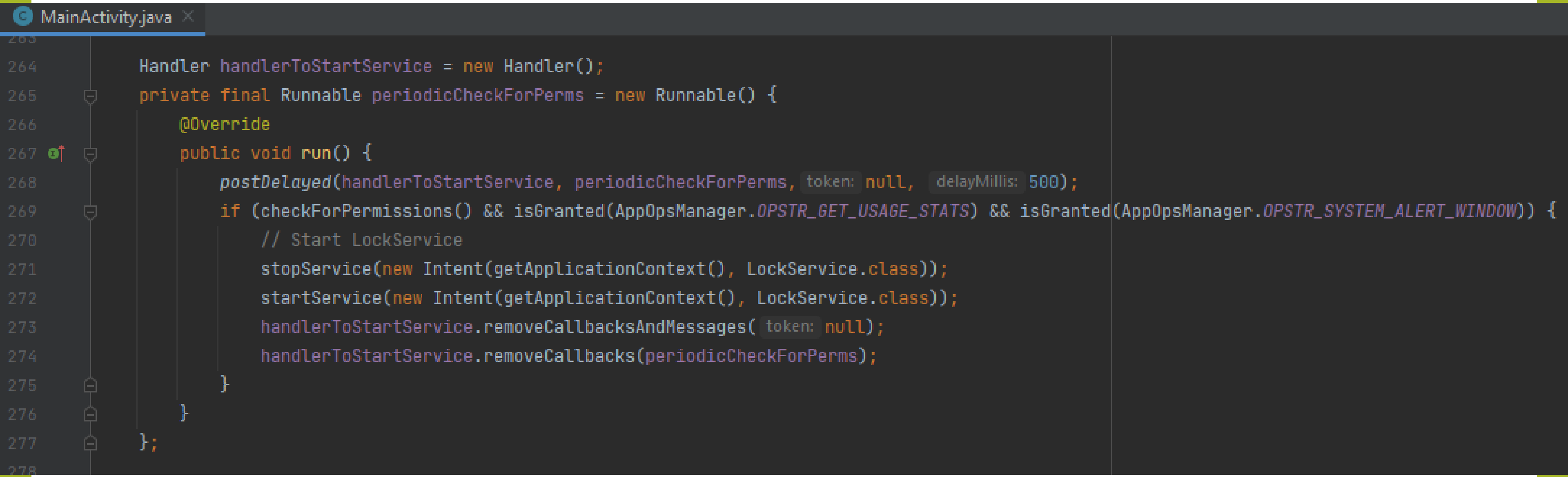
```java
    private Boolean checkForPermissions() {
        // Check if physical activity and write storage permissions are granted
        for (String perm : new String[] {"android.permission.ACTIVITY_RECOGNITION","android.permission.WRITE_EXTERNAL_STORAGE"} ){
            if (ContextCompat.checkSelfPermission( this,perm) != PackageManager.PERMISSION_GRANTED) {
                return false;
            }
        }
        return true;
    }

    // Callback after permissions requested
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults){
        // Check if permissions are granted
        if (!checkForPermissions()) {
            openAlertDialog();
        }
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    }
```



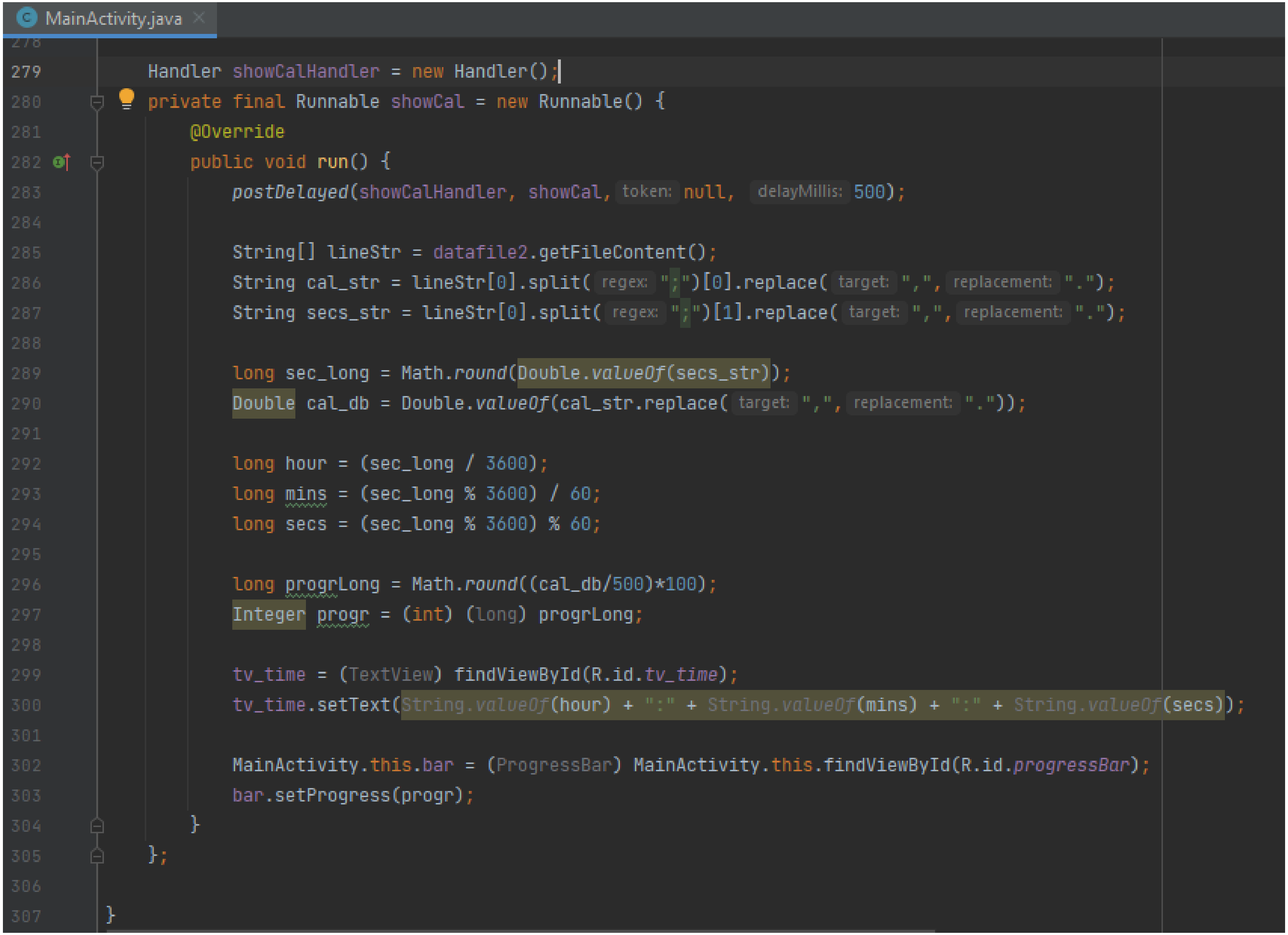
```java
    // Dialog in case of permission are not granted
    private void openAlertDialog() {
        AlertDialog.Builder adb = new AlertDialog.Builder( context: this);
        adb.setMessage("Cette application nécessite l'accès aux contenus multimédias et aux données relatives à l'activité physique");
        adb.setPositiveButton( text: "OK",
                new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // Stop app if permissions are not granted
                        dialog.dismiss();
                        finish();
                    }
                });
        AlertDialog ad = adb.create();
        ad.show();
    }
```



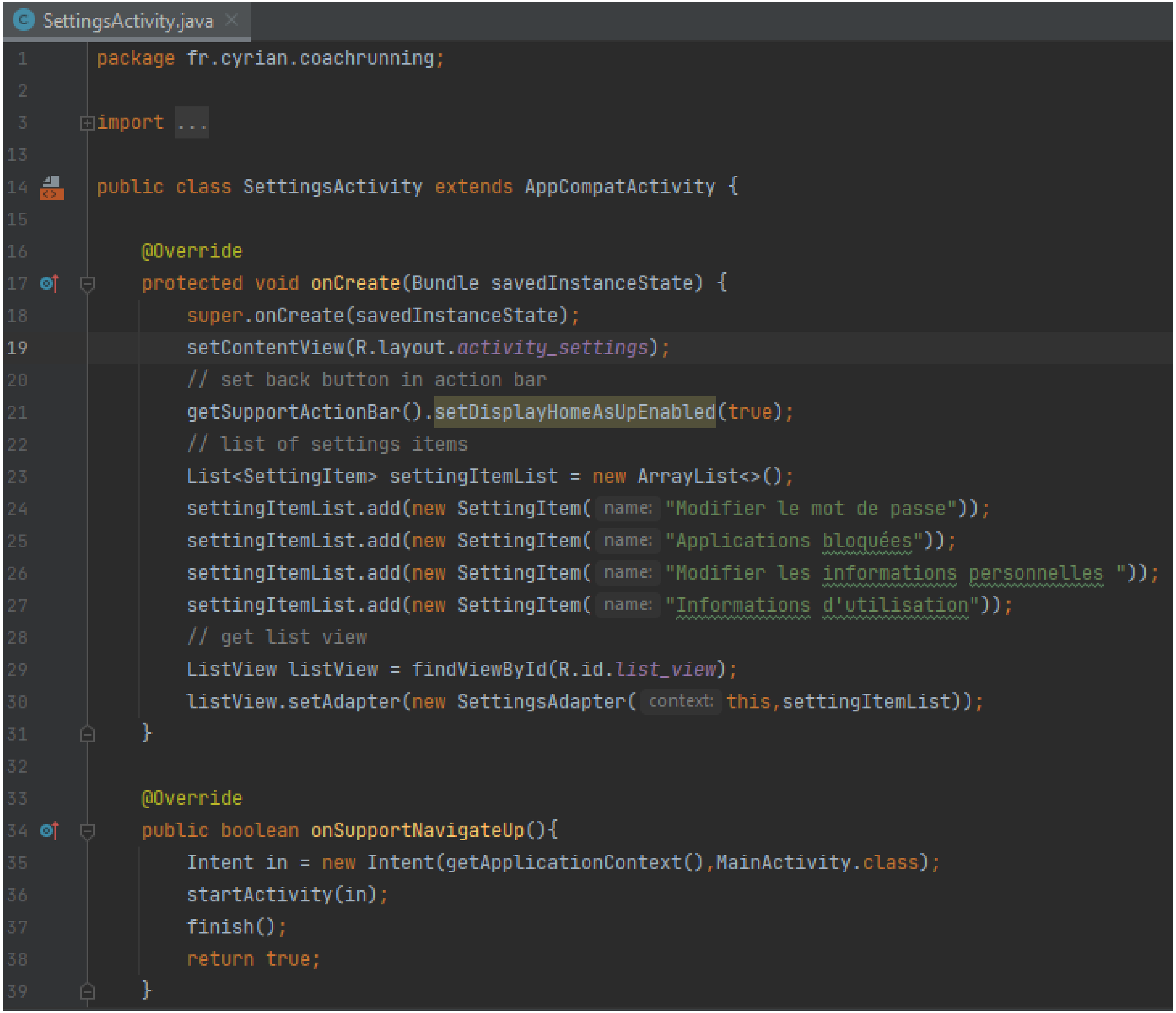
```java
    // Dialog that send to specific special permission and finish app
    public void askForSpecialPerms(String action, String message) {
        AlertDialog ad = new AlertDialog.Builder( context: MainActivity.this).create();
        ad.setTitle("Permission needed");
        ad.setMessage(message);
        ad.setButton(AlertDialog.BUTTON_NEUTRAL,  text: "OK",
                new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                        Intent i = new Intent(action);
                        startActivity(i);
                        finish();
                    }
                });
        ad.show();
    }

    // Check if special permission has been granted for app
    public boolean isGranted(String op) {
        AppOpsManager appOps = (AppOpsManager) getApplicationContext().getSystemService(Context.APP_OPS_SERVICE);
        int mode = appOps.checkOpNoThrow(op, android.os.Process.myUid(), getApplicationContext().getPackageName());
        boolean granted = (mode == AppOpsManager.MODE_ALLOWED);
        return granted;
    }

```



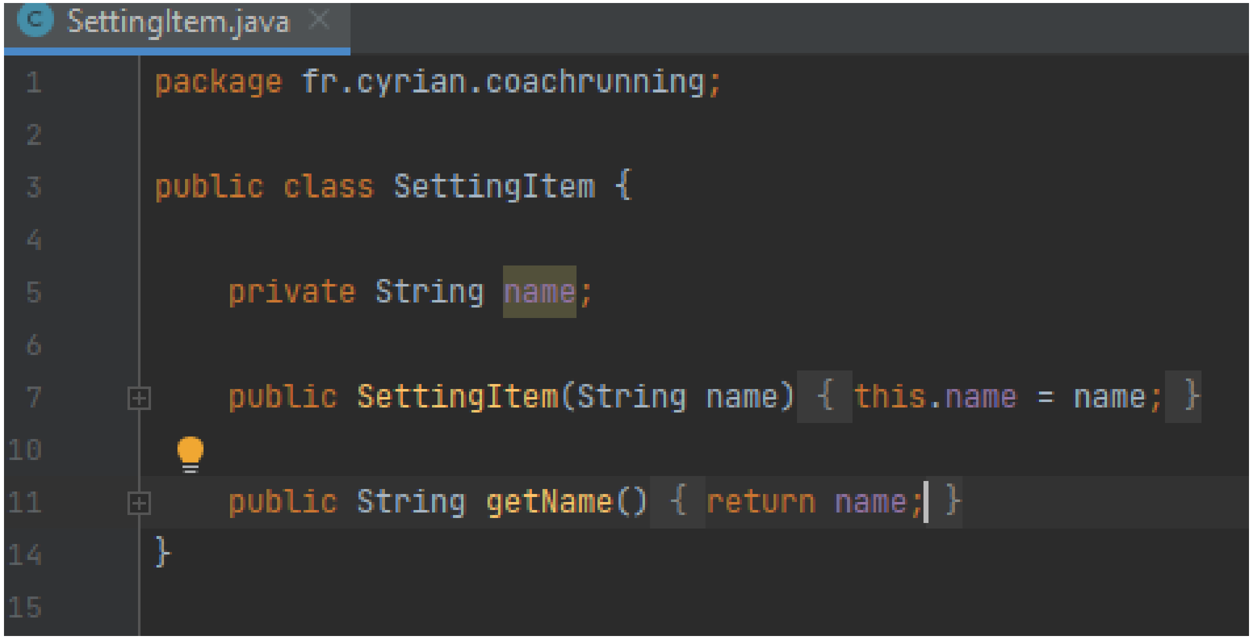
```java
    Handler handlerToStartService = new Handler();
    private final Runnable periodicCheckForPerms = new Runnable() {
        @Override
        public void run() {
            postDelayed(handlerToStartService, periodicCheckForPerms, token: null, delayMillis: 500);
            if (checkForPermissions() && isGranted(AppOpsManager.OPSTR_GET_USAGE_STATS) && isGranted(AppOpsManager.OPSTR_SYSTEM_ALERT_WINDOW)) {
                // Start LockService
                stopService(new Intent(getApplicationContext(), LockService.class));
                startService(new Intent(getApplicationContext(), LockService.class));
                handlerToStartService.removeCallbacksAndMessages( token: null);
                handlerToStartService.removeCallbacks(periodicCheckForPerms);
            }
        }
    };
```



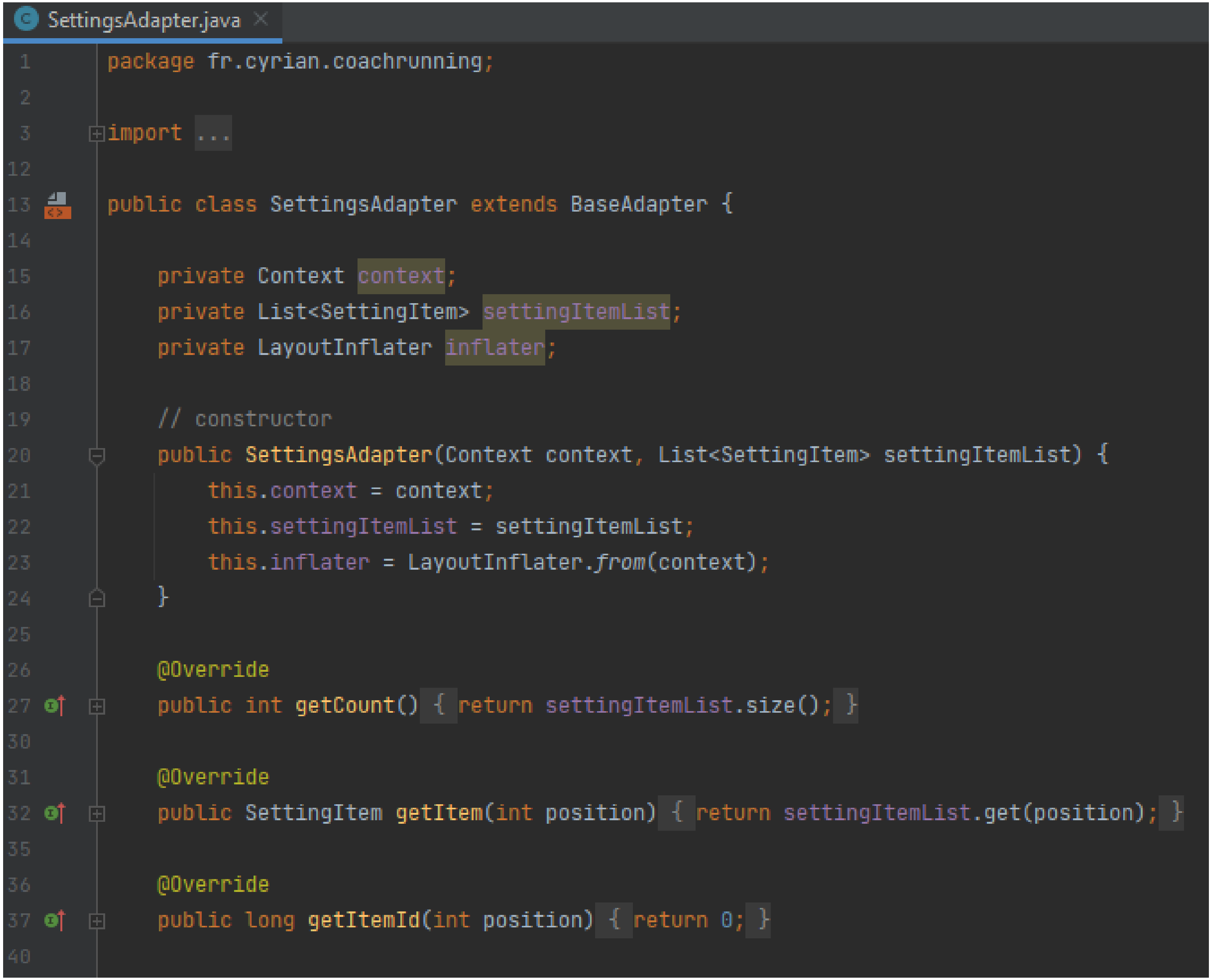
```java
    Handler showCalHandler = new Handler();
    private final Runnable showCal = new Runnable() {
        @Override
        public void run() {
            postDelayed(showCalHandler, showCal, token: null,  delayMillis: 500);

            String[] lineStr = datafile2.getFileContent();
            String cal_str = lineStr[0].split( regex: ";")[0].replace( target: ",", replacement: ".");
            String secs_str = lineStr[0].split( regex: ";")[1].replace( target: ",", replacement: ".");

            long sec_long = Math.round(Double.valueOf(secs_str));
            Double cal_db = Double.valueOf(cal_str.replace( target: ",", replacement: "."));

            long hour = (sec_long / 3600);
            long mins = (sec_long % 3600) / 60;
            long secs = (sec_long % 3600) % 60;

            long progrLong = Math.round((cal_db/500)*100);
            Integer progr = (int) (long) progrLong;

            tv_time = (TextView) findViewById(R.id.tv_time);
            tv_time.setText(String.valueOf(hour) + ":" + String.valueOf(mins) + ":" + String.valueOf(secs));

            MainActivity.this.bar = (ProgressBar) MainActivity.this.findViewById(R.id.progressBar);
            bar.setProgress(progr);
        }
    };

}
```



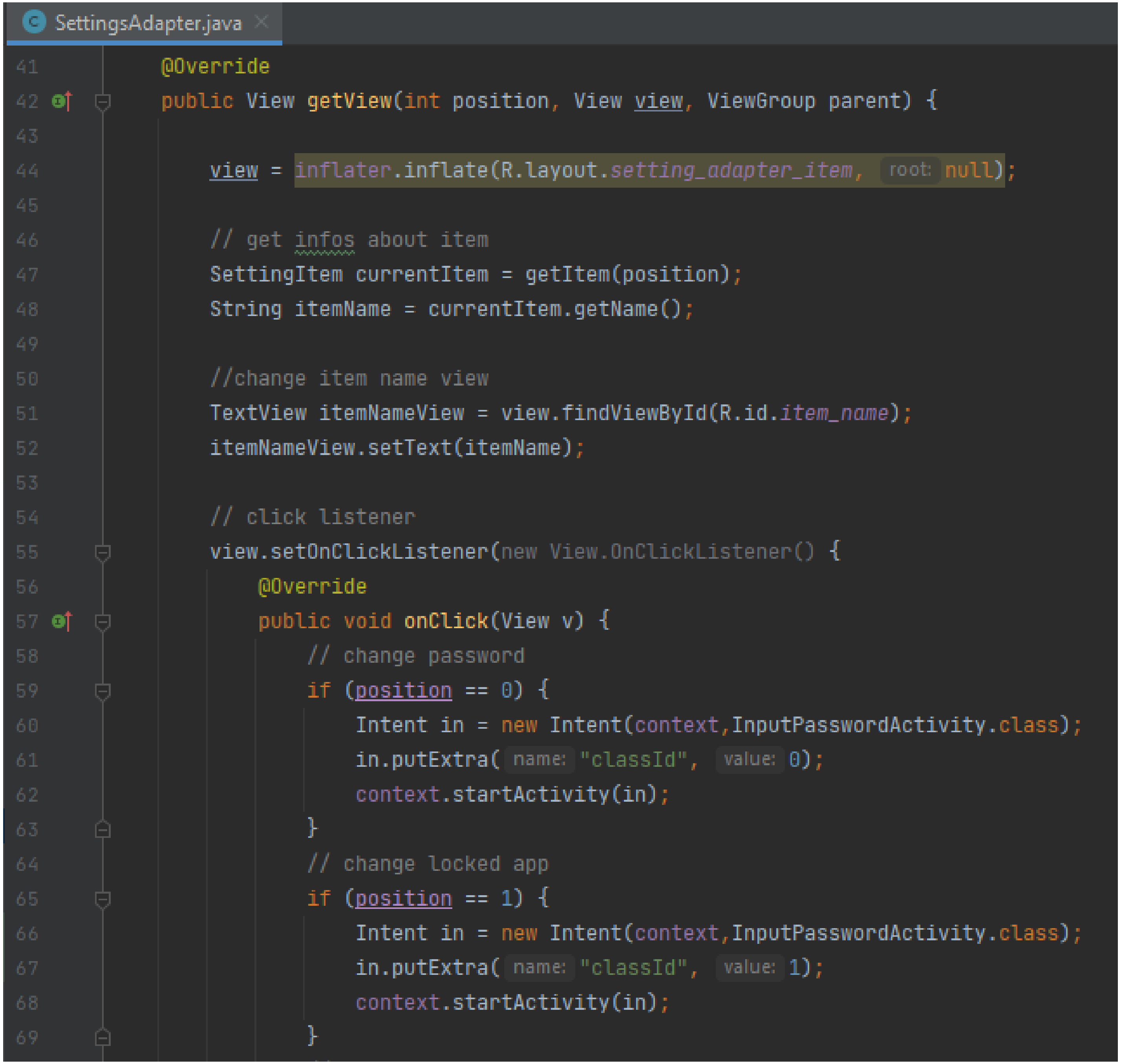
```java
package fr.cyrian.coachrunning;

import ...

public class SettingsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);
        // set back button in action bar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        // list of settings items
        List<SettingItem> settingItemList = new ArrayList<>();
        settingItemList.add(new SettingItem( name: "Modifier le mot de passe"));
        settingItemList.add(new SettingItem( name: "Applications bloquées"));
        settingItemList.add(new SettingItem( name: "Modifier les informations personnelles "));
        settingItemList.add(new SettingItem( name: "Informations d'utilisation"));
        // get list view
        ListView listView = findViewById(R.id.list_view);
        listView.setAdapter(new SettingsAdapter( context: this,settingItemList));
    }

    @Override
    public boolean onSupportNavigateUp(){
        Intent in = new Intent(getApplicationContext(),MainActivity.class);
        startActivity(in);
        finish();
        return true;
    }
```



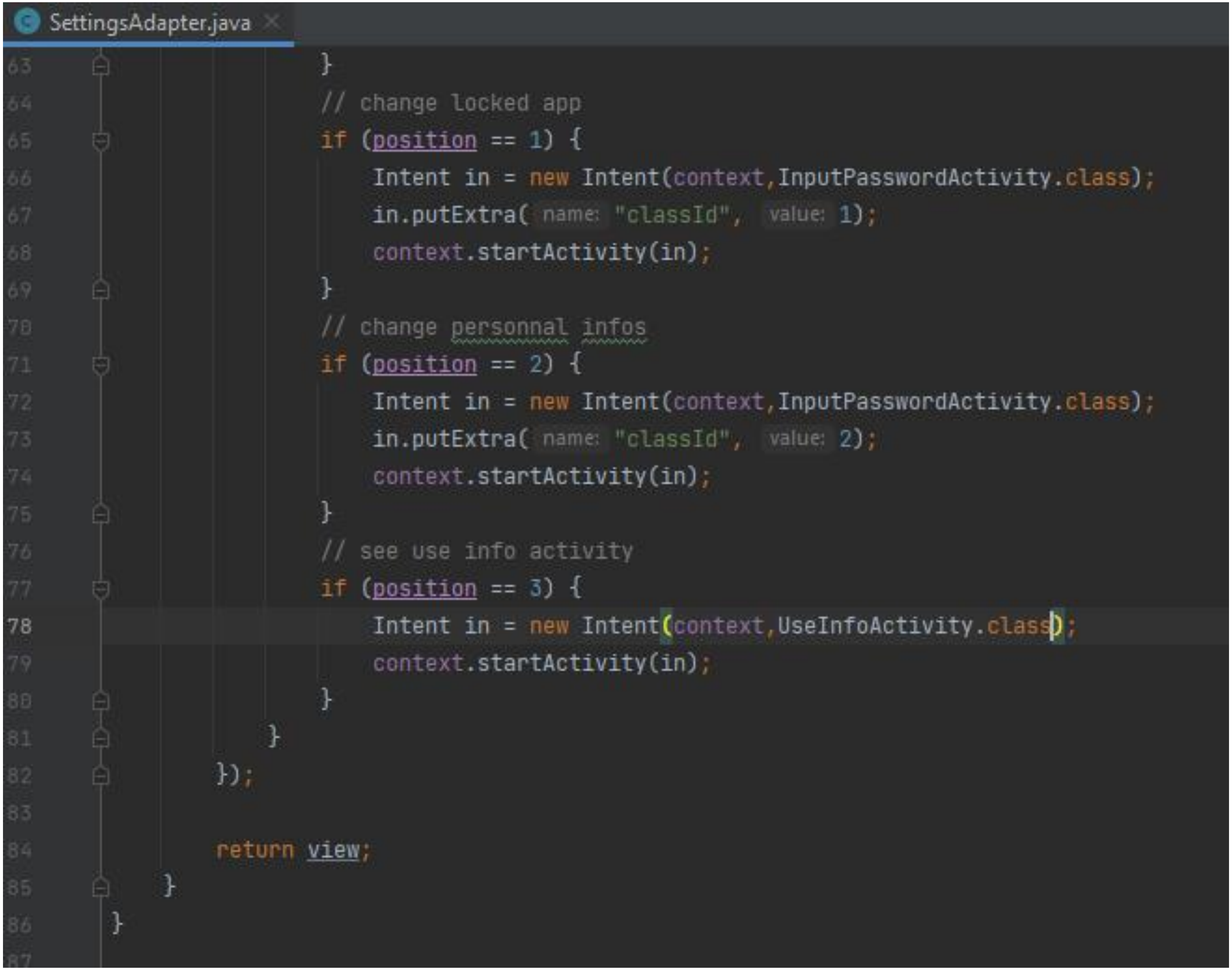
```java
package fr.cyrian.coachrunning;

public class SettingItem {

    private String name;

    public SettingItem(String name) { this.name = name; }

    public String getName() { return name; }
}

```



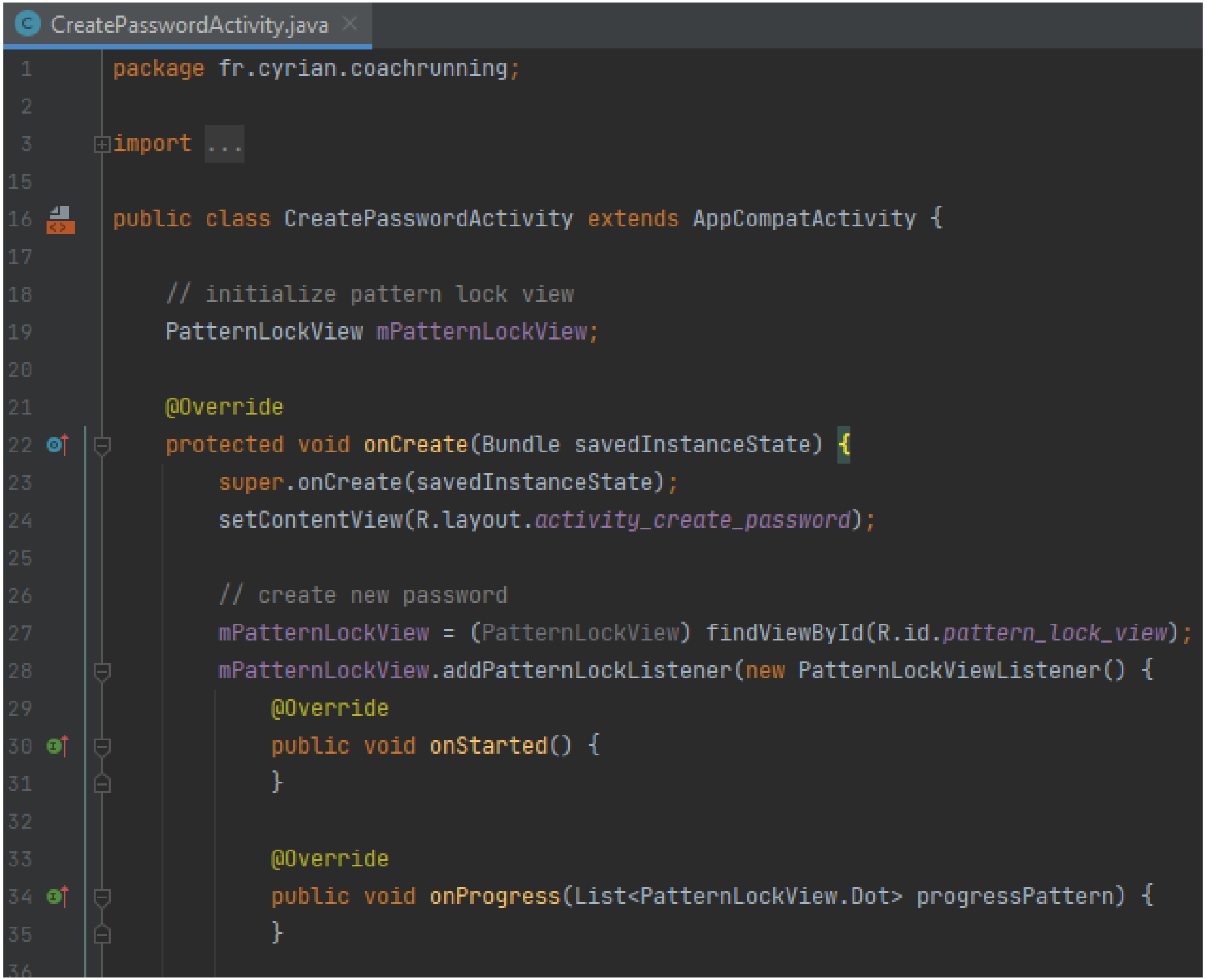
```java
package fr.cyrian.coachrunning;

import ...

public class SettingsAdapter extends BaseAdapter {

    private Context context;
    private List<SettingItem> settingItemList;
    private LayoutInflater inflater;

    // constructor
    public SettingsAdapter(Context context, List<SettingItem> settingItemList) {
        this.context = context;
        this.settingItemList = settingItemList;
        this.inflater = LayoutInflater.from(context);
    }

    @Override
    public int getCount() { return settingItemList.size(); }

    @Override
    public SettingItem getItem(int position) { return settingItemList.get(position); }

    @Override
    public long getItemId(int position) { return 0; }

```



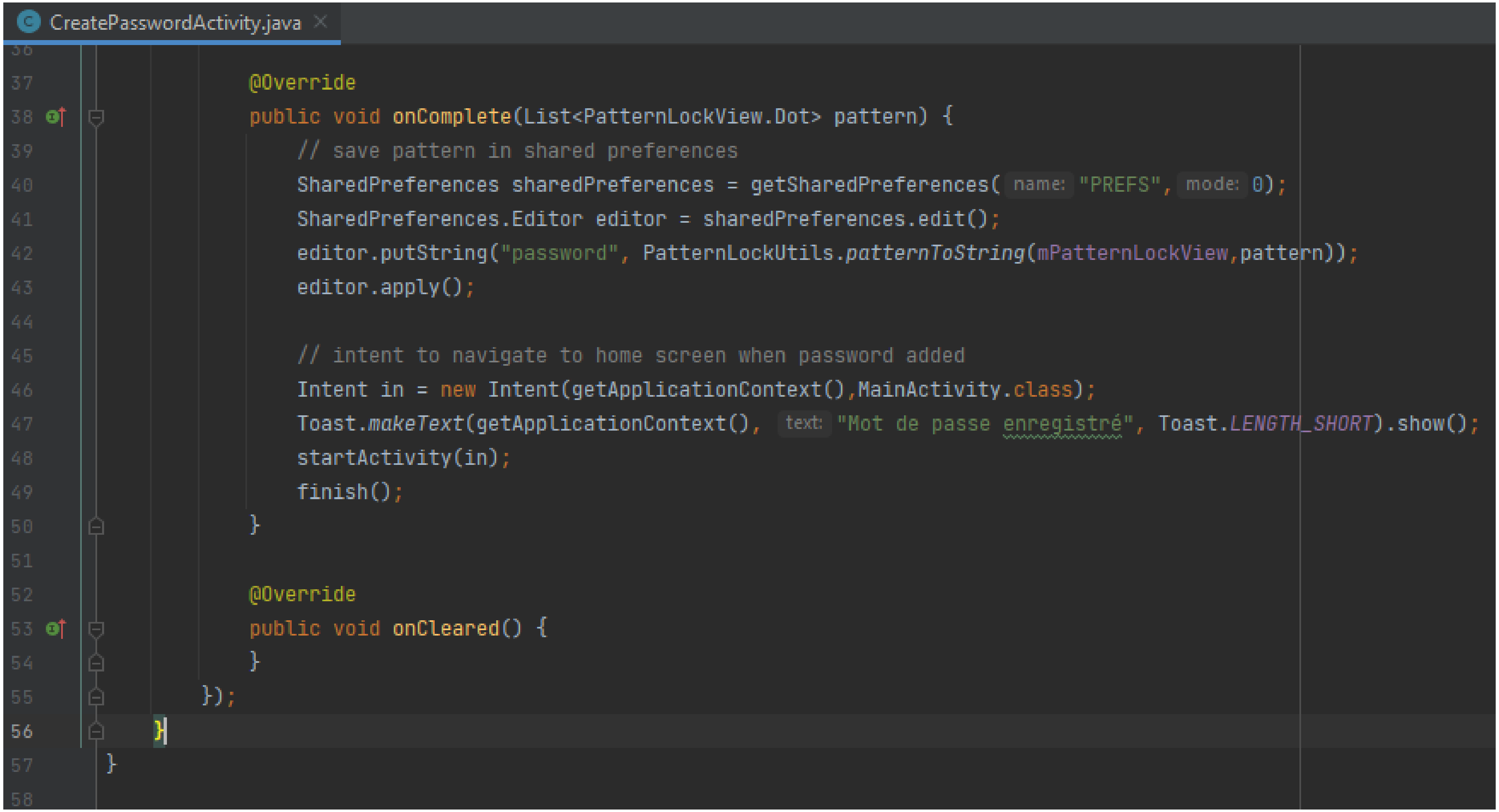
```java
@Override
public View getView(int position, View view, ViewGroup parent) {

    view = inflater.inflate(R.layout.setting_adapter_item,  root: null);

    // get infos about item
    SettingItem currentItem = getItem(position);
    String itemName = currentItem.getName();

    //change item name view
    TextView itemNameView = view.findViewById(R.id.item_name);
    itemNameView.setText(itemName);

    // click listener
    view.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // change password
            if (position == 0) {
                Intent in = new Intent(context,InputPasswordActivity.class);
                in.putExtra( name: "classId",  value: 0);
                context.startActivity(in);
            }
            // change locked app
            if (position == 1) {
                Intent in = new Intent(context,InputPasswordActivity.class);
                in.putExtra( name: "classId",  value: 1);
                context.startActivity(in);
            }
```



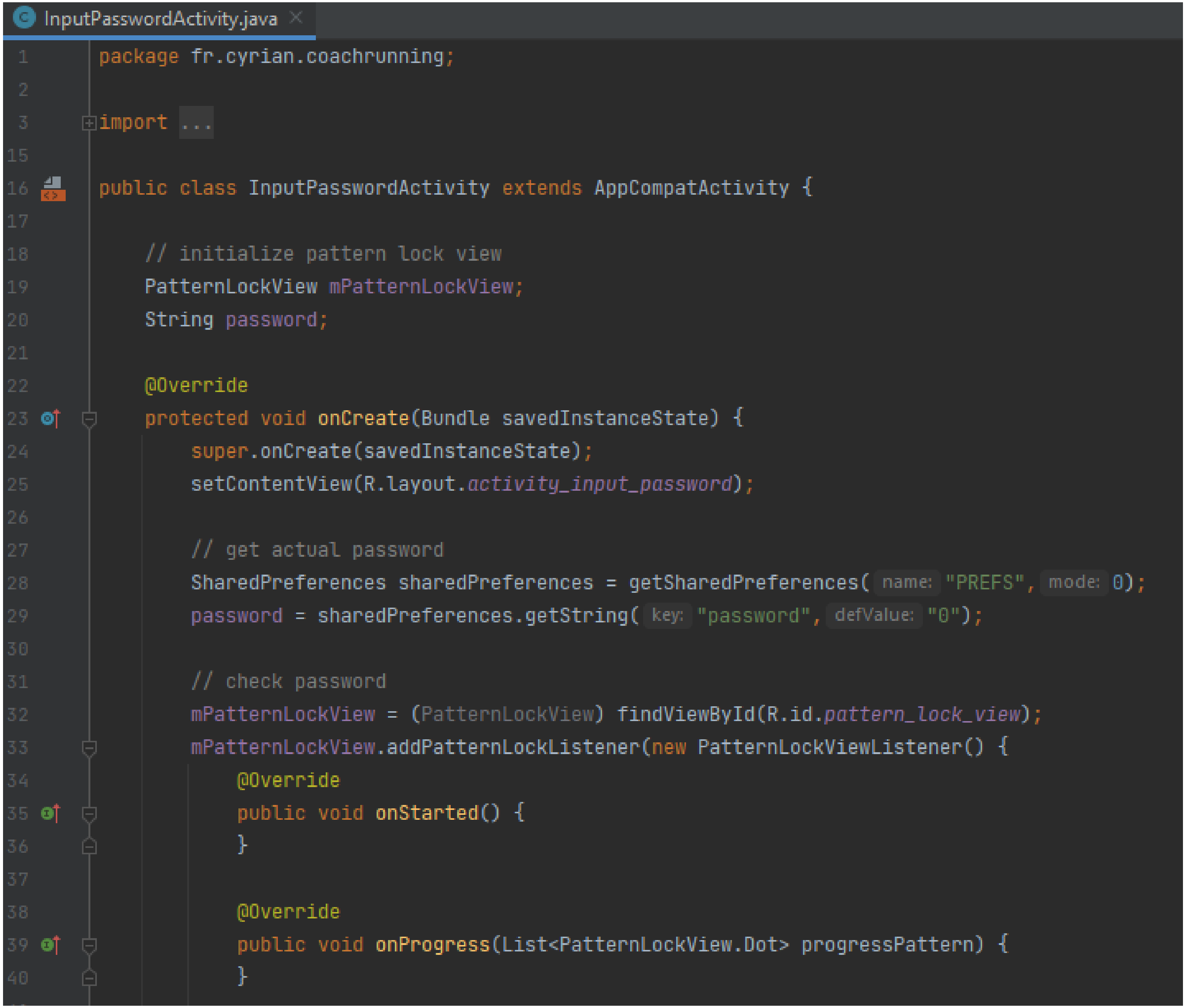
```
                }
                // change locked app
                if (position == 1) {
                    Intent in = new Intent(context,InputPasswordActivity.class);
                    in.putExtra( name: "classId",  value: 1);
                    context.startActivity(in);
                }
                // change personnal infos
                if (position == 2) {
                    Intent in = new Intent(context,InputPasswordActivity.class);
                    in.putExtra( name: "classId",  value: 2);
                    context.startActivity(in);
                }
                // see use info activity
                if (position == 3) {
                    Intent in = new Intent(context,UseInfoActivity.class);
                    context.startActivity(in);
                }
            }
        });

        return view;
    }
}
```



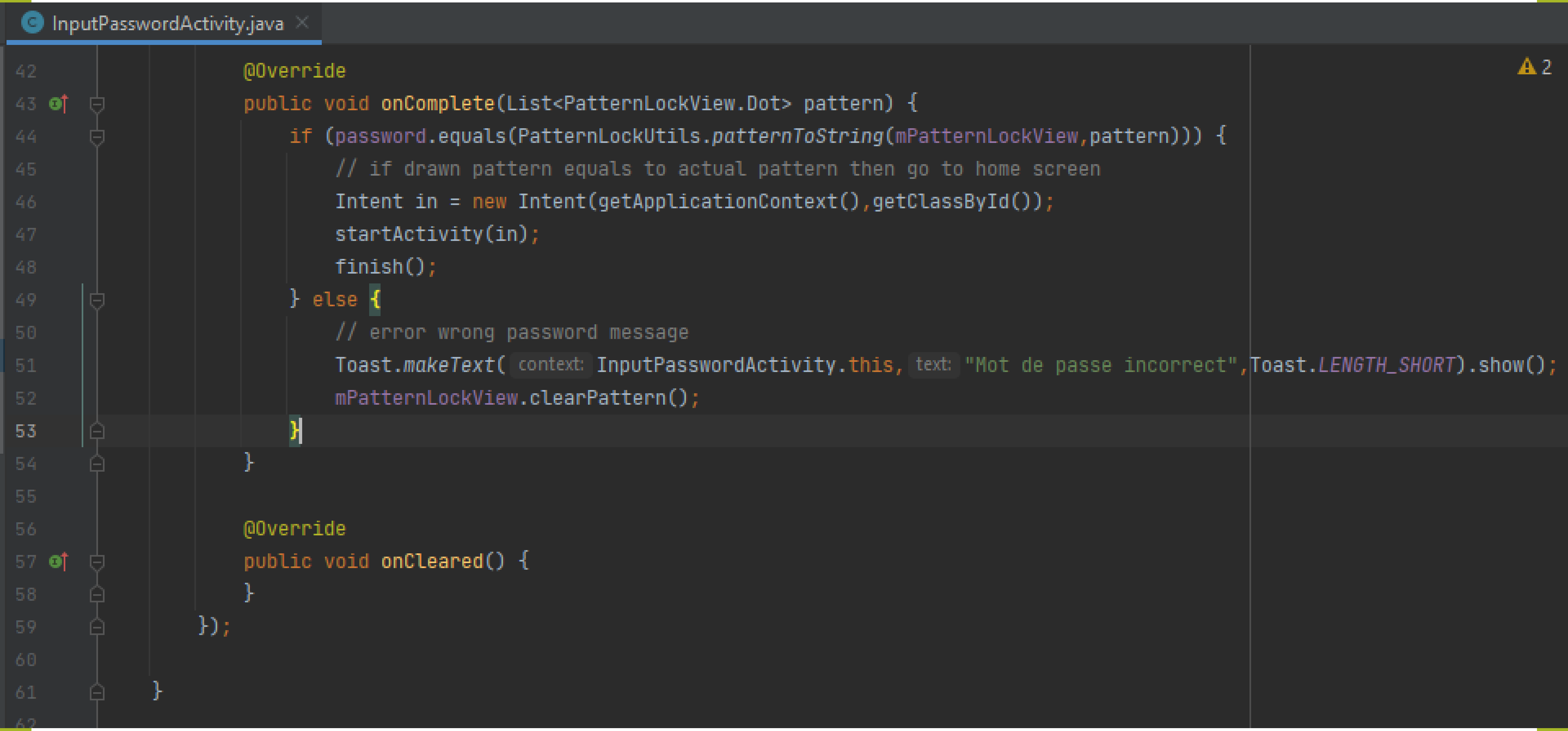
```
package fr.cyrian.coachrunning;

import ...

public class CreatePasswordActivity extends AppCompatActivity {

    // initialize pattern lock view
    PatternLockView mPatternLockView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_password);

        // create new password
        mPatternLockView = (PatternLockView) findViewById(R.id.pattern_lock_view);
        mPatternLockView.addPatternLockListener(new PatternLockViewListener() {
            @Override
            public void onStarted() {
            }

            @Override
            public void onProgress(List<PatternLockView.Dot> progressPattern) {
            }
```



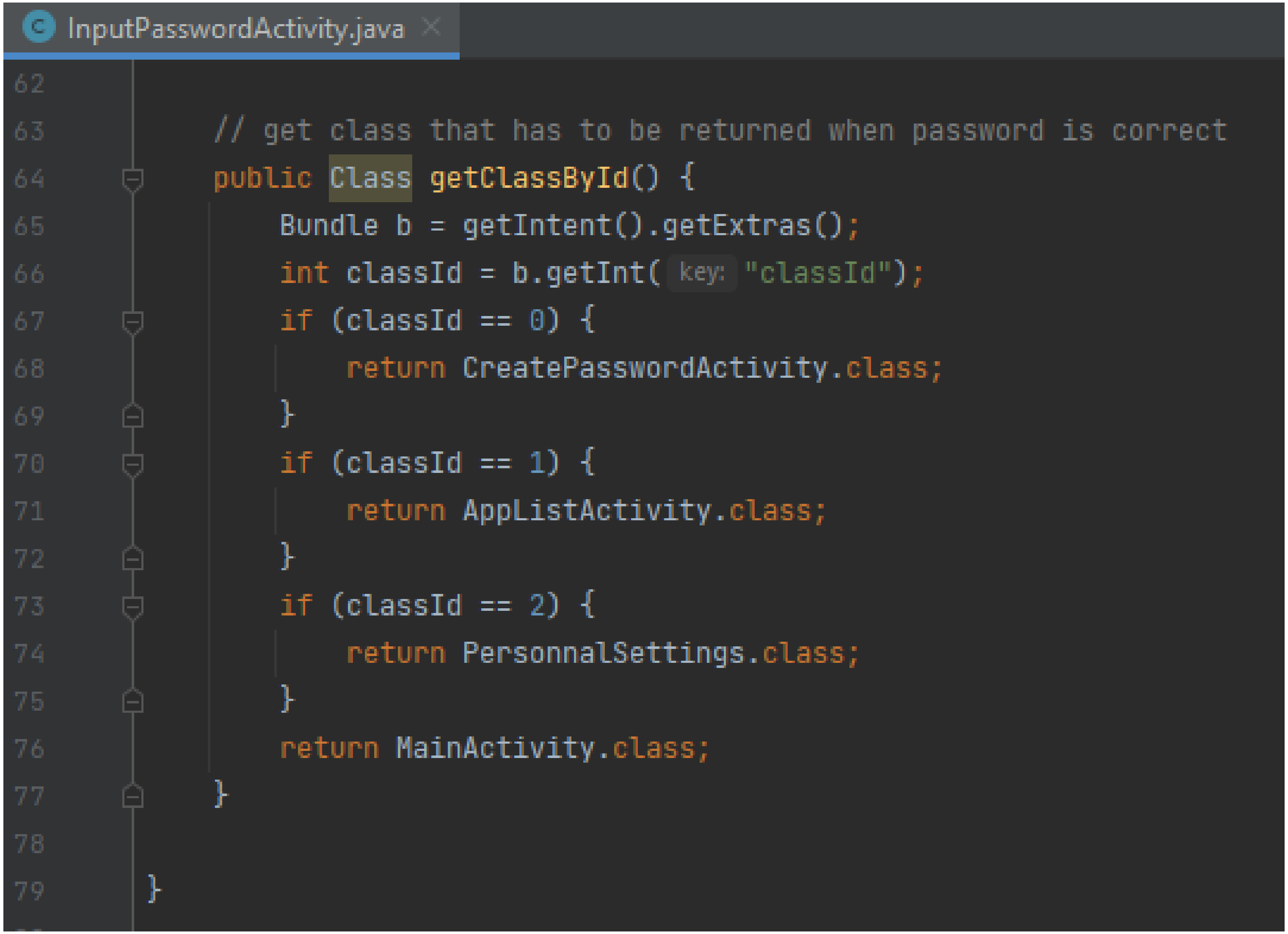
```java
            @Override
            public void onComplete(List<PatternLockView.Dot> pattern) {
                // save pattern in shared preferences
                SharedPreferences sharedPreferences = getSharedPreferences( name: "PREFS", mode: 0);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putString("password", PatternLockUtils.patternToString(mPatternLockView,pattern));
                editor.apply();

                // intent to navigate to home screen when password added
                Intent in = new Intent(getApplicationContext(),MainActivity.class);
                Toast.makeText(getApplicationContext(), text: "Mot de passe enregistré", Toast.LENGTH_SHORT).show();
                startActivity(in);
                finish();
            }

            @Override
            public void onCleared() {
            }
        });
    }
}
```



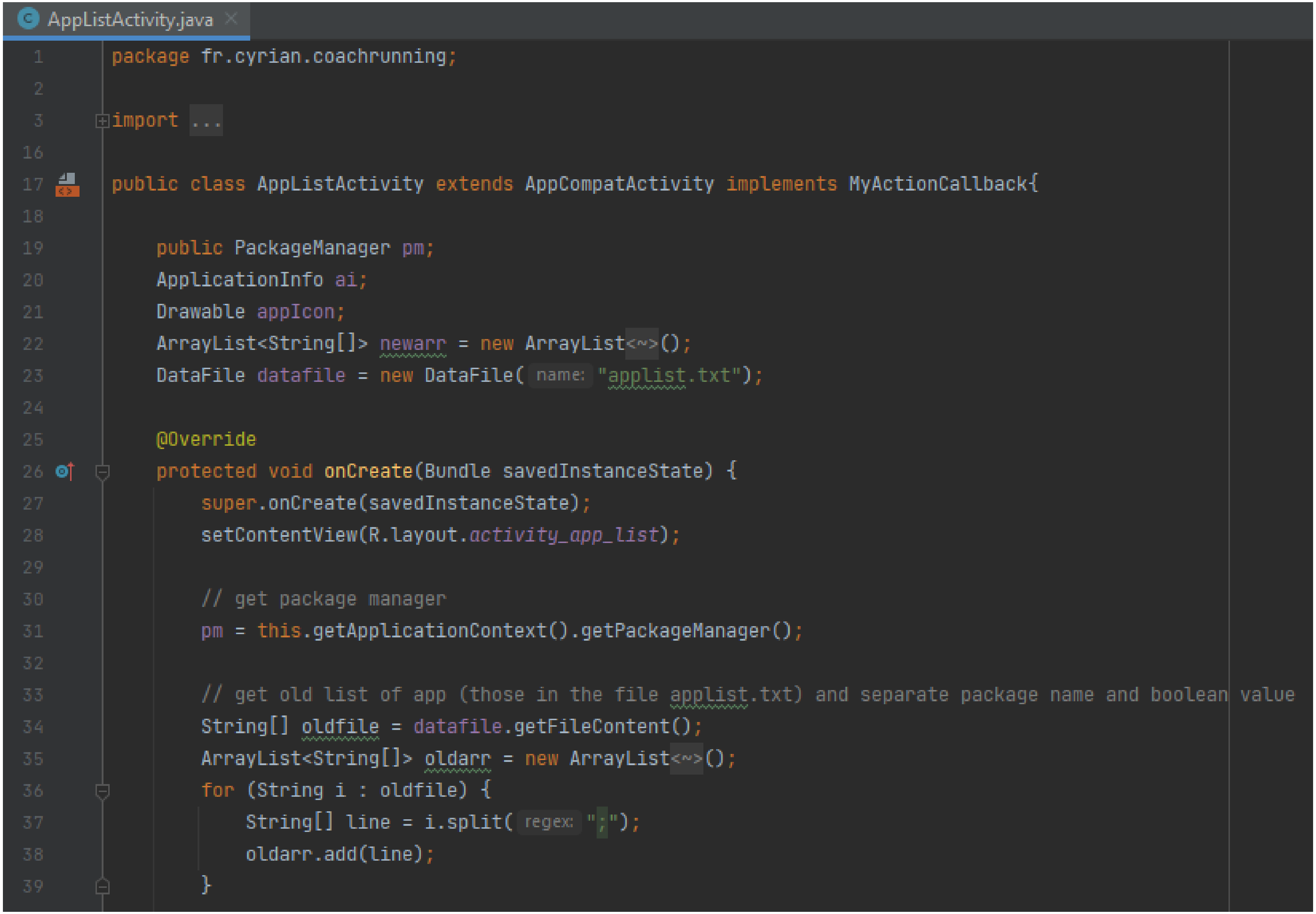
```java
package fr.cyrian.coachrunning;

import ...

public class InputPasswordActivity extends AppCompatActivity {

    // initialize pattern lock view
    PatternLockView mPatternLockView;
    String password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input_password);

        // get actual password
        SharedPreferences sharedPreferences = getSharedPreferences( name: "PREFS", mode: 0);
        password = sharedPreferences.getString( key: "password", defValue: "0");

        // check password
        mPatternLockView = (PatternLockView) findViewById(R.id.pattern_lock_view);
        mPatternLockView.addPatternLockListener(new PatternLockViewListener() {
            @Override
            public void onStarted() {
            }

            @Override
            public void onProgress(List<PatternLockView.Dot> progressPattern) {
            }
```



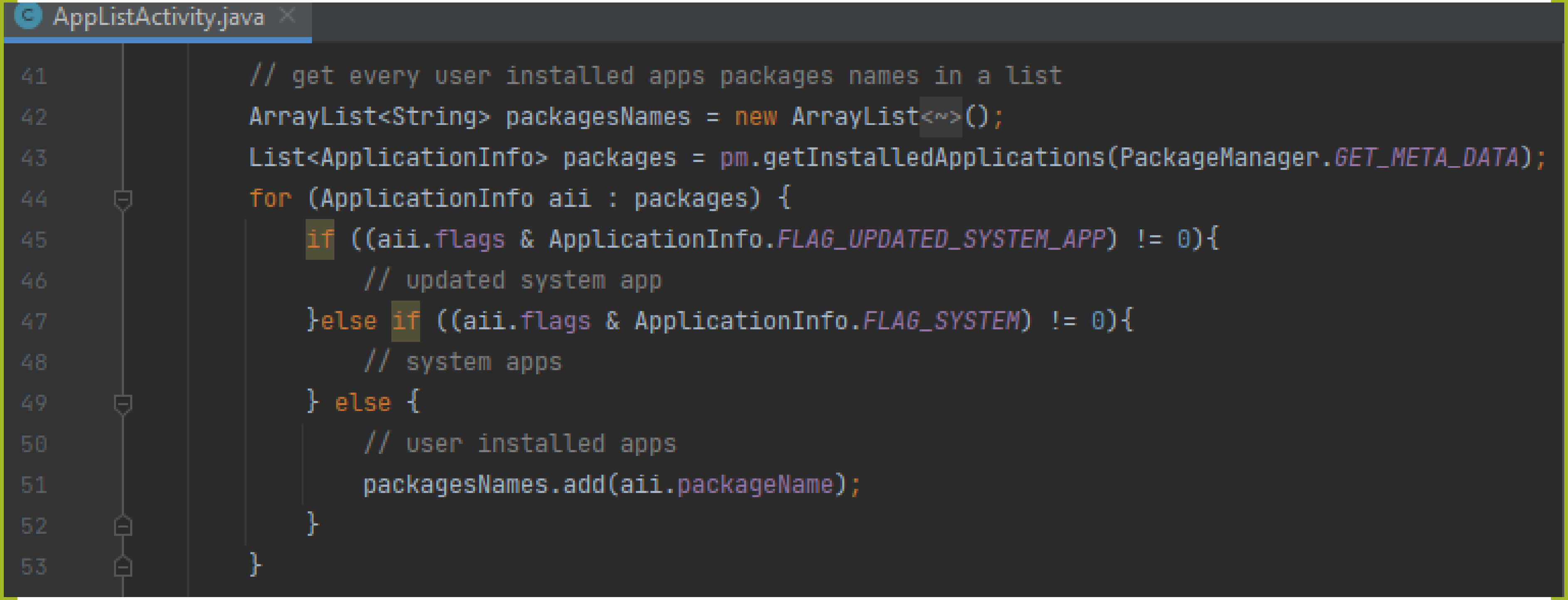
```
            @Override
            public void onComplete(List<PatternLockView.Dot> pattern) {
                if (password.equals(PatternLockUtils.patternToString(mPatternLockView,pattern))) {
                    // if drawn pattern equals to actual pattern then go to home screen
                    Intent in = new Intent(getApplicationContext(),getClassById());
                    startActivity(in);
                    finish();
                } else {
                    // error wrong password message
                    Toast.makeText( context: InputPasswordActivity.this, text: "Mot de passe incorrect",Toast.LENGTH_SHORT).show();
                    mPatternLockView.clearPattern();
                }
            }

            @Override
            public void onCleared() {
            }
        });

    }
```



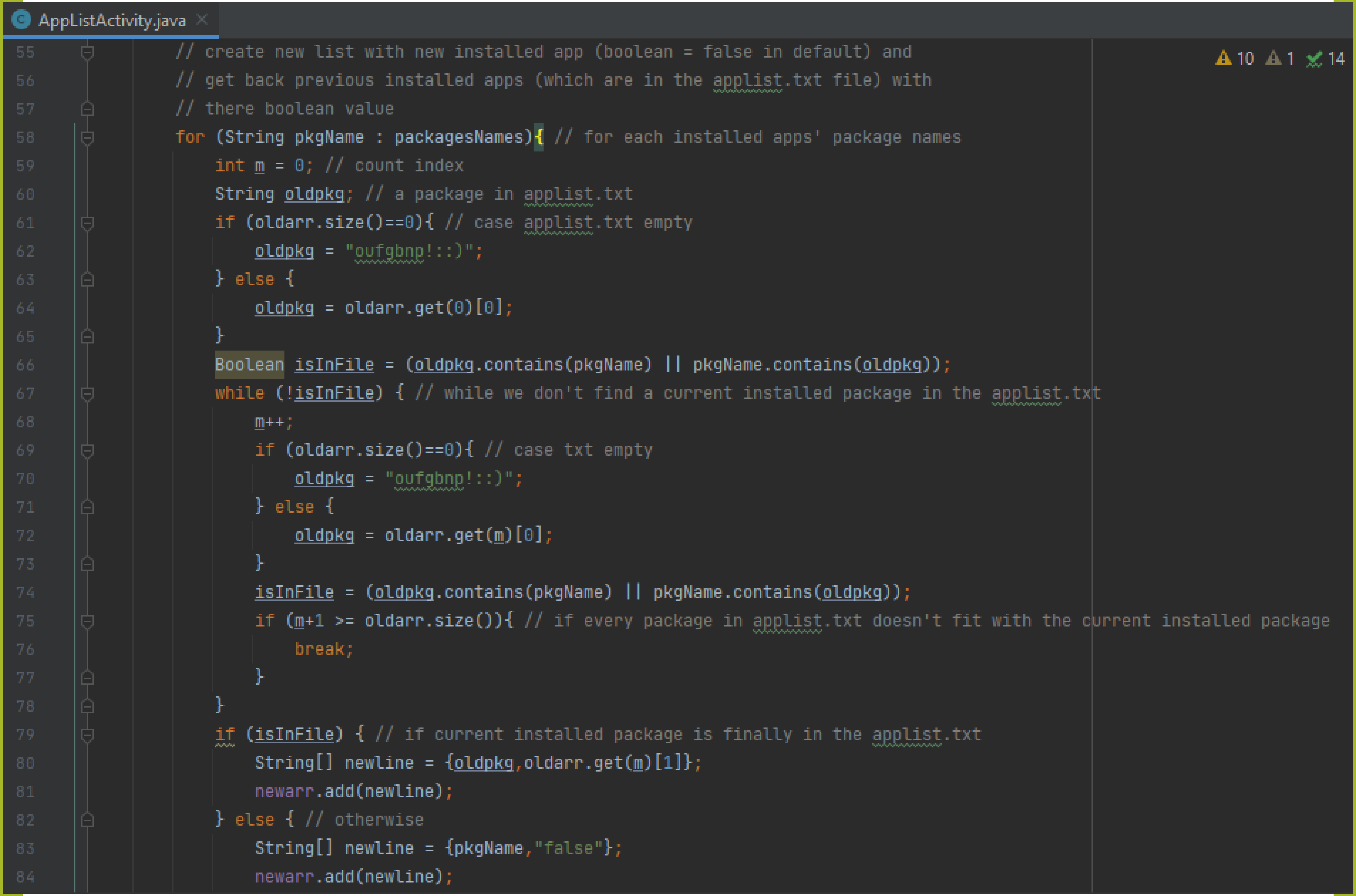
```
    // get class that has to be returned when password is correct
    public Class getClassById() {
        Bundle b = getIntent().getExtras();
        int classId = b.getInt( key: "classId");
        if (classId == 0) {
            return CreatePasswordActivity.class;
        }
        if (classId == 1) {
            return AppListActivity.class;
        }
        if (classId == 2) {
            return PersonnalSettings.class;
        }
        return MainActivity.class;
    }

}
```



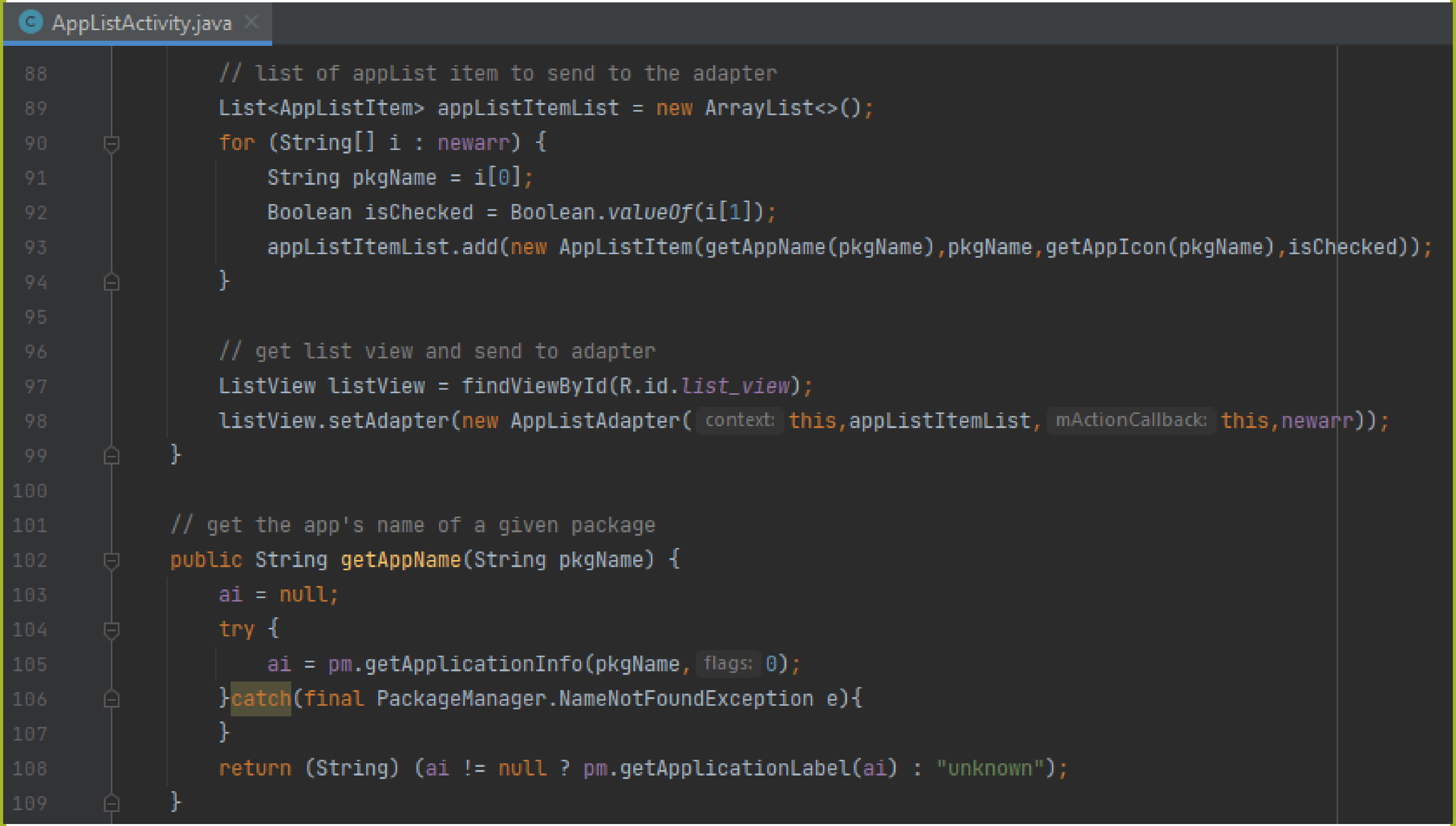
```java
package fr.cyrian.coachrunning;

import ...

public class AppListActivity extends AppCompatActivity implements MyActionCallback{

    public PackageManager pm;
    ApplicationInfo ai;
    Drawable appIcon;
    ArrayList<String[]> newarr = new ArrayList<~>();
    DataFile datafile = new DataFile( name: "applist.txt");

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_app_list);

        // get package manager
        pm = this.getApplicationContext().getPackageManager();

        // get old list of app (those in the file applist.txt) and separate package name and boolean value
        String[] oldfile = datafile.getFileContent();
        ArrayList<String[]> oldarr = new ArrayList<~>();
        for (String i : oldfile) {
            String[] line = i.split( regex: ";");
            oldarr.add(line);
        }
```



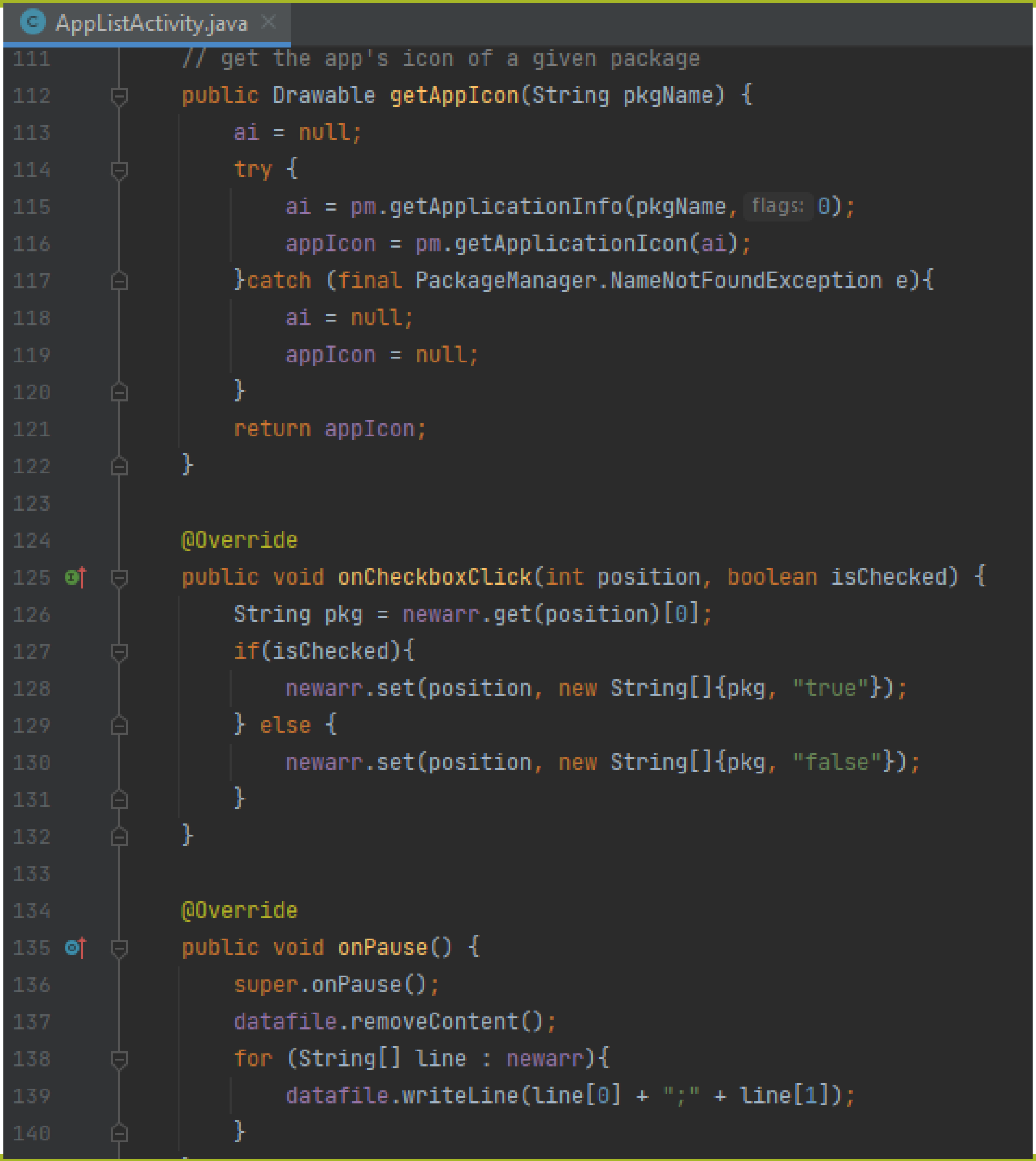
```java
        // get every user installed apps packages names in a list
        ArrayList<String> packagesNames = new ArrayList<~>();
        List<ApplicationInfo> packages = pm.getInstalledApplications(PackageManager.GET_META_DATA);
        for (ApplicationInfo aii : packages) {
            if ((aii.flags & ApplicationInfo.FLAG_UPDATED_SYSTEM_APP) != 0){
                // updated system app
            }else if ((aii.flags & ApplicationInfo.FLAG_SYSTEM) != 0){
                // system apps
            } else {
                // user installed apps
                packagesNames.add(aii.packageName);
            }
        }
```



```java
        // create new list with new installed app (boolean = false in default) and
        // get back previous installed apps (which are in the applist.txt file) with
        // there boolean value
        for (String pkgName : packagesNames){ // for each installed apps' package names
            int m = 0; // count index
            String oldpkg; // a package in applist.txt
            if (oldarr.size()==0){ // case applist.txt empty
                oldpkg = "oufgbnp!::)";
            } else {
                oldpkg = oldarr.get(0)[0];
            }
            Boolean isInFile = (oldpkg.contains(pkgName) || pkgName.contains(oldpkg));
            while (!isInFile) { // while we don't find a current installed package in the applist.txt
                m++;
                if (oldarr.size()==0){ // case txt empty
                    oldpkg = "oufgbnp!::)";
                } else {
                    oldpkg = oldarr.get(m)[0];
                }
                isInFile = (oldpkg.contains(pkgName) || pkgName.contains(oldpkg));
                if (m+1 >= oldarr.size()){ // if every package in applist.txt doesn't fit with the current installed package
                    break;
                }
            }
            if (isInFile) { // if current installed package is finally in the applist.txt
                String[] newline = {oldpkg,oldarr.get(m)[1]};
                newarr.add(newline);
            } else { // otherwise
                String[] newline = {pkgName,"false"};
                newarr.add(newline);
```

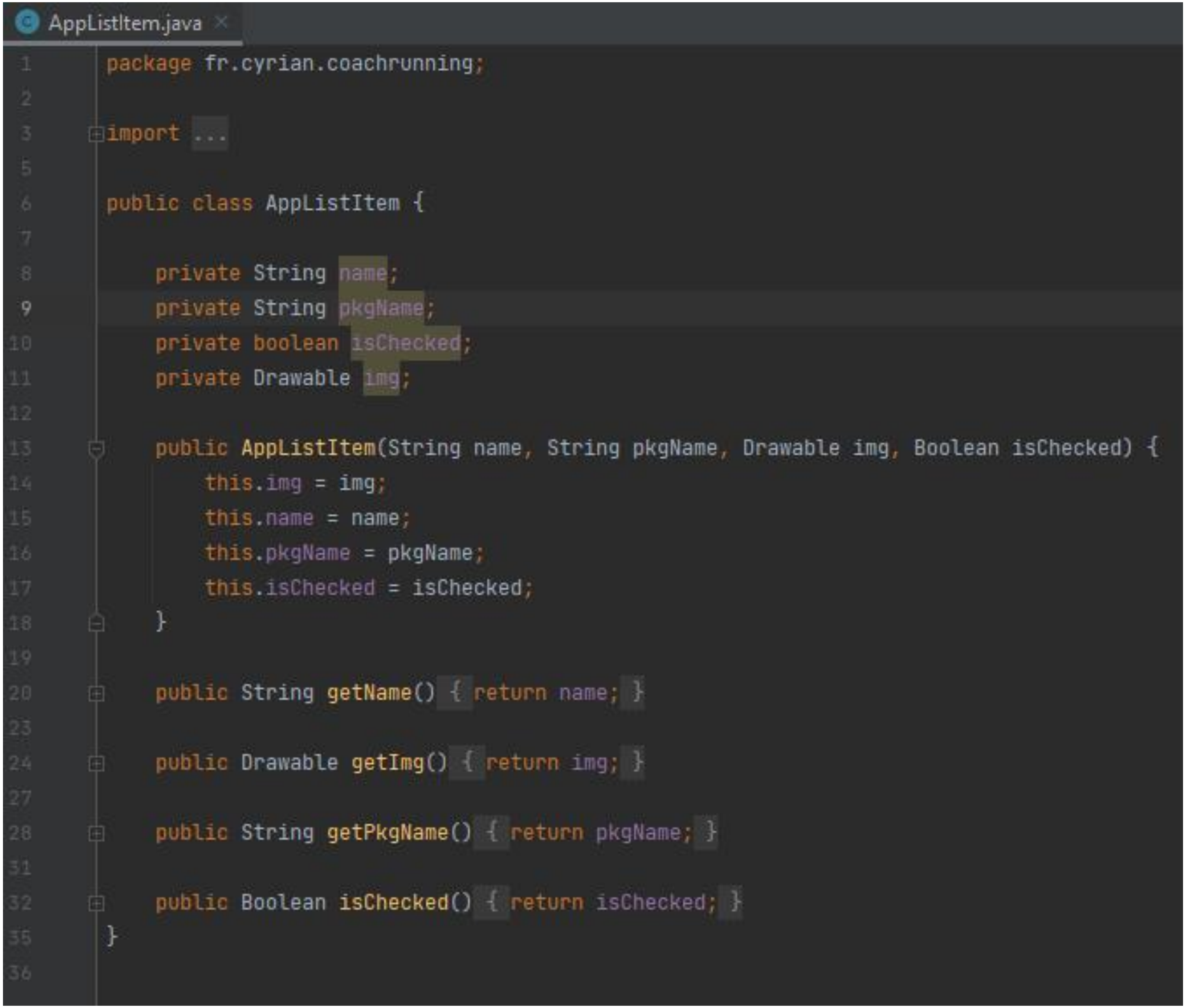


```java
        // list of appList item to send to the adapter
        List<AppListItem> appListItemList = new ArrayList<>();
        for (String[] i : newarr) {
            String pkgName = i[0];
            Boolean isChecked = Boolean.valueOf(i[1]);
            appListItemList.add(new AppListItem(getAppName(pkgName),pkgName,getAppIcon(pkgName),isChecked));
        }

        // get list view and send to adapter
        ListView listView = findViewById(R.id.list_view);
        listView.setAdapter(new AppListAdapter( context: this,appListItemList, mActionCallback: this,newarr));
    }

    // get the app's name of a given package
    public String getAppName(String pkgName) {
        ai = null;
        try {
            ai = pm.getApplicationInfo(pkgName, flags: 0);
        }catch(final PackageManager.NameNotFoundException e){
        }
        return (String) (ai != null ? pm.getApplicationLabel(ai) : "unknown");
    }
```

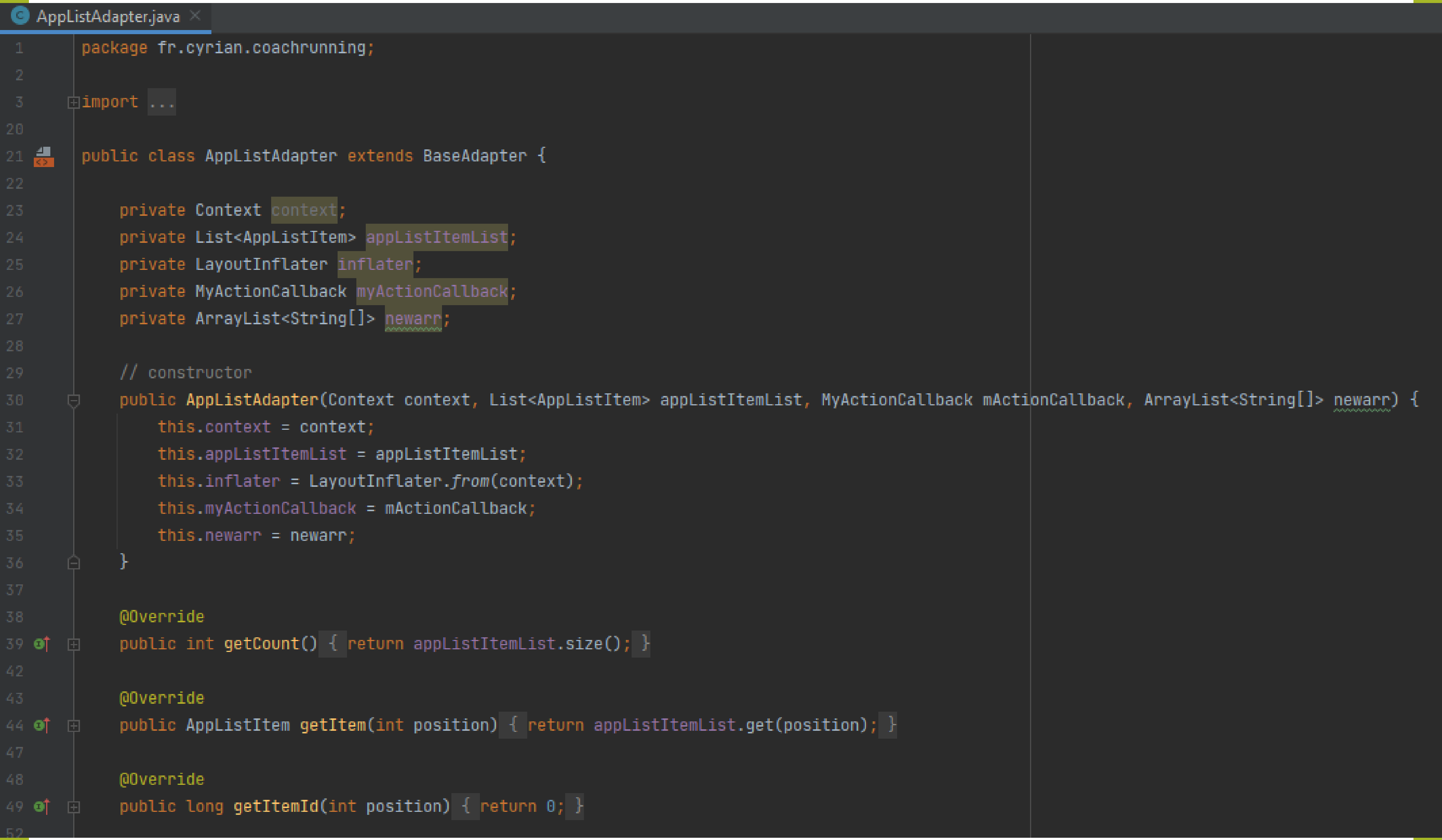


AppListActivity.java

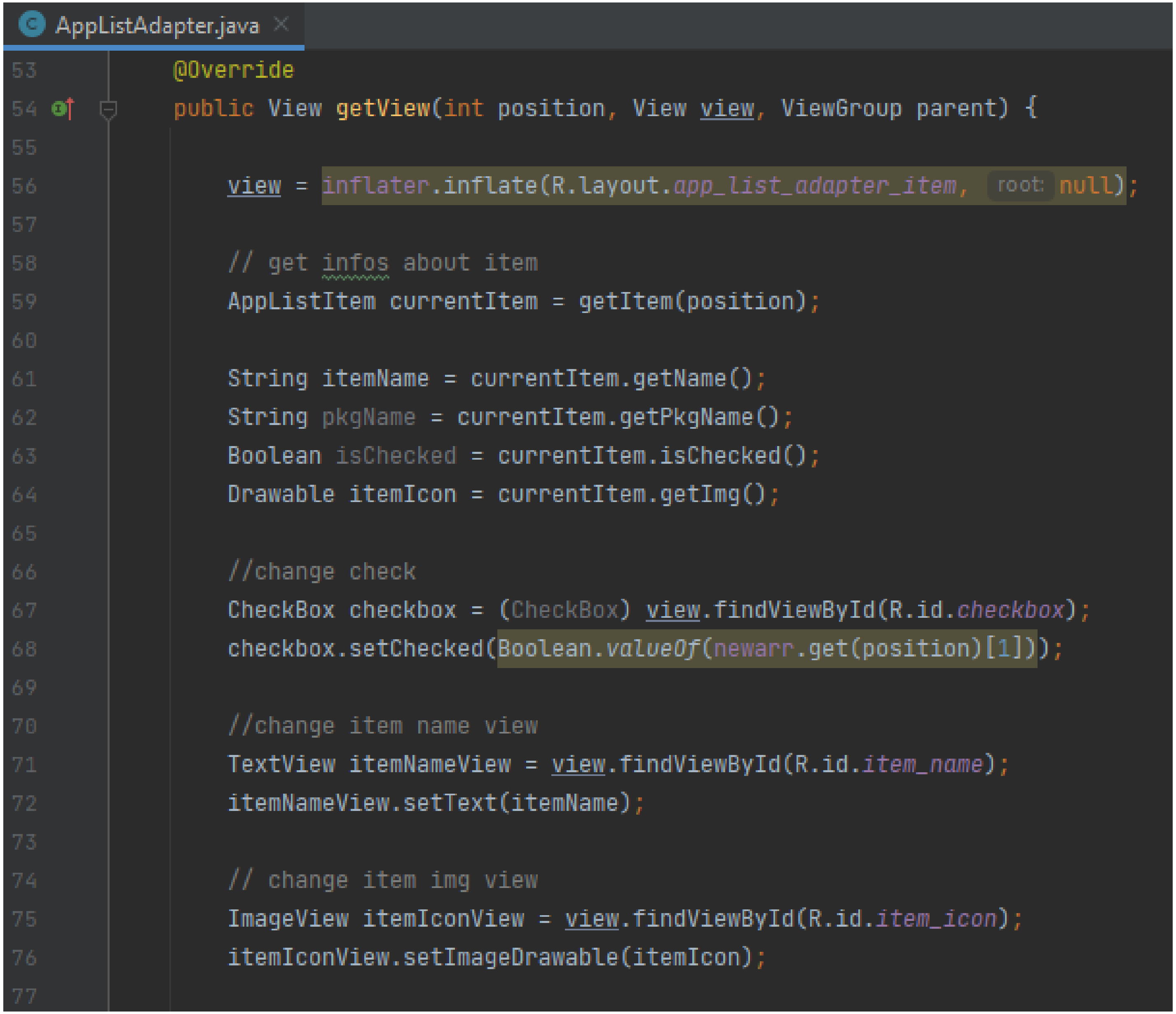
```java
    // get the app's icon of a given package
    public Drawable getAppIcon(String pkgName) {
        ai = null;
        try {
            ai = pm.getApplicationInfo(pkgName, flags: 0);
            appIcon = pm.getApplicationIcon(ai);
        }catch (final PackageManager.NameNotFoundException e){
            ai = null;
            appIcon = null;
        }
        return appIcon;
    }

    @Override
    public void onCheckboxClick(int position, boolean isChecked) {
        String pkg = newarr.get(position)[0];
        if(isChecked){
            newarr.set(position, new String[]{pkg, "true"});
        } else {
            newarr.set(position, new String[]{pkg, "false"});
        }
    }

    @Override
    public void onPause() {
        super.onPause();
        datafile.removeContent();
        for (String[] line : newarr){
            datafile.writeLine(line[0] + ";" + line[1]);
        }
```



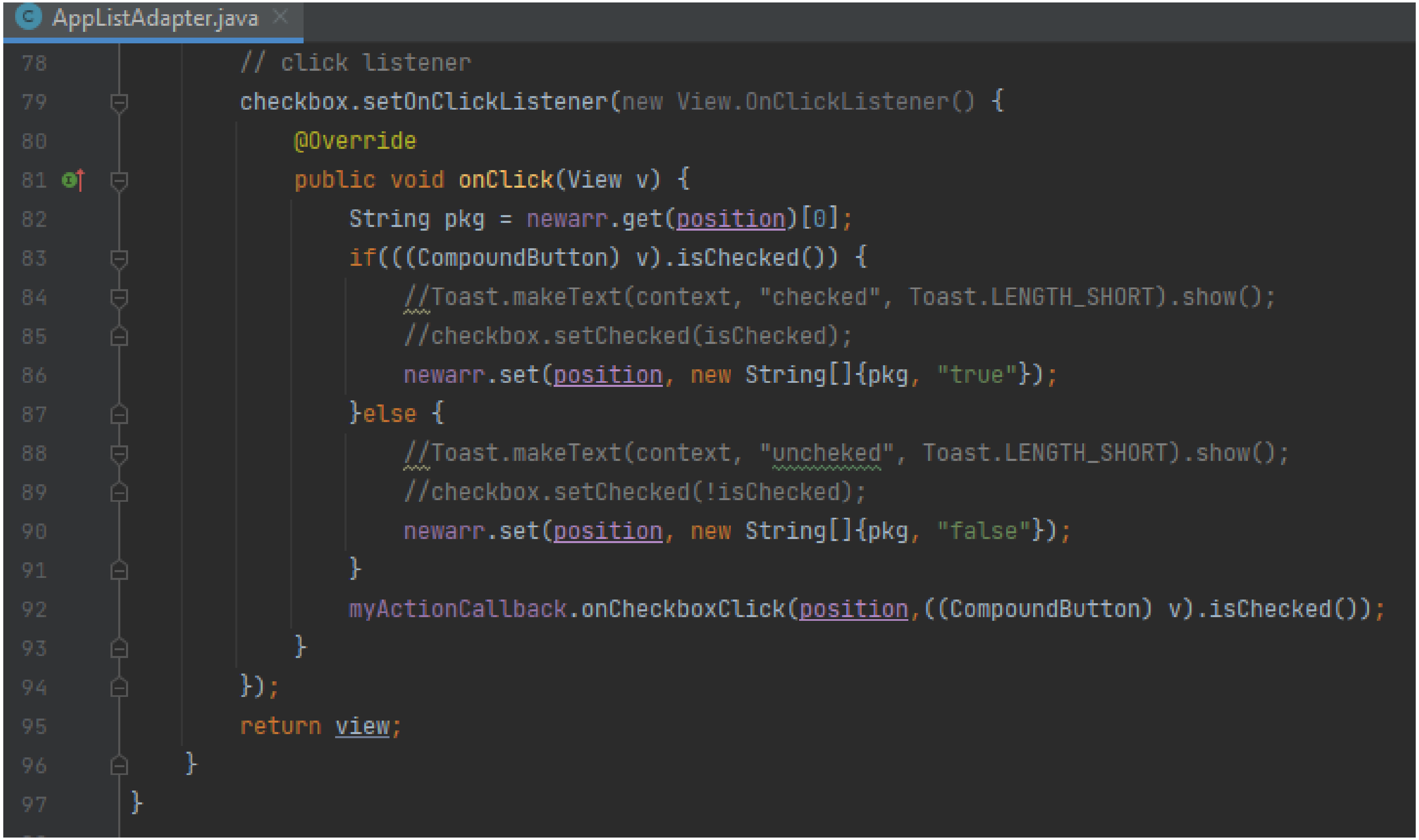
```java
package fr.cyrian.coachrunning;

import ...

public class AppListItem {

    private String name;
    private String pkgName;
    private boolean isChecked;
    private Drawable img;

    public AppListItem(String name, String pkgName, Drawable img, Boolean isChecked) {
        this.img = img;
        this.name = name;
        this.pkgName = pkgName;
        this.isChecked = isChecked;
    }

    public String getName() { return name; }

    public Drawable getImg() { return img; }

    public String getPkgName() { return pkgName; }

    public Boolean isChecked() { return isChecked; }
}

```



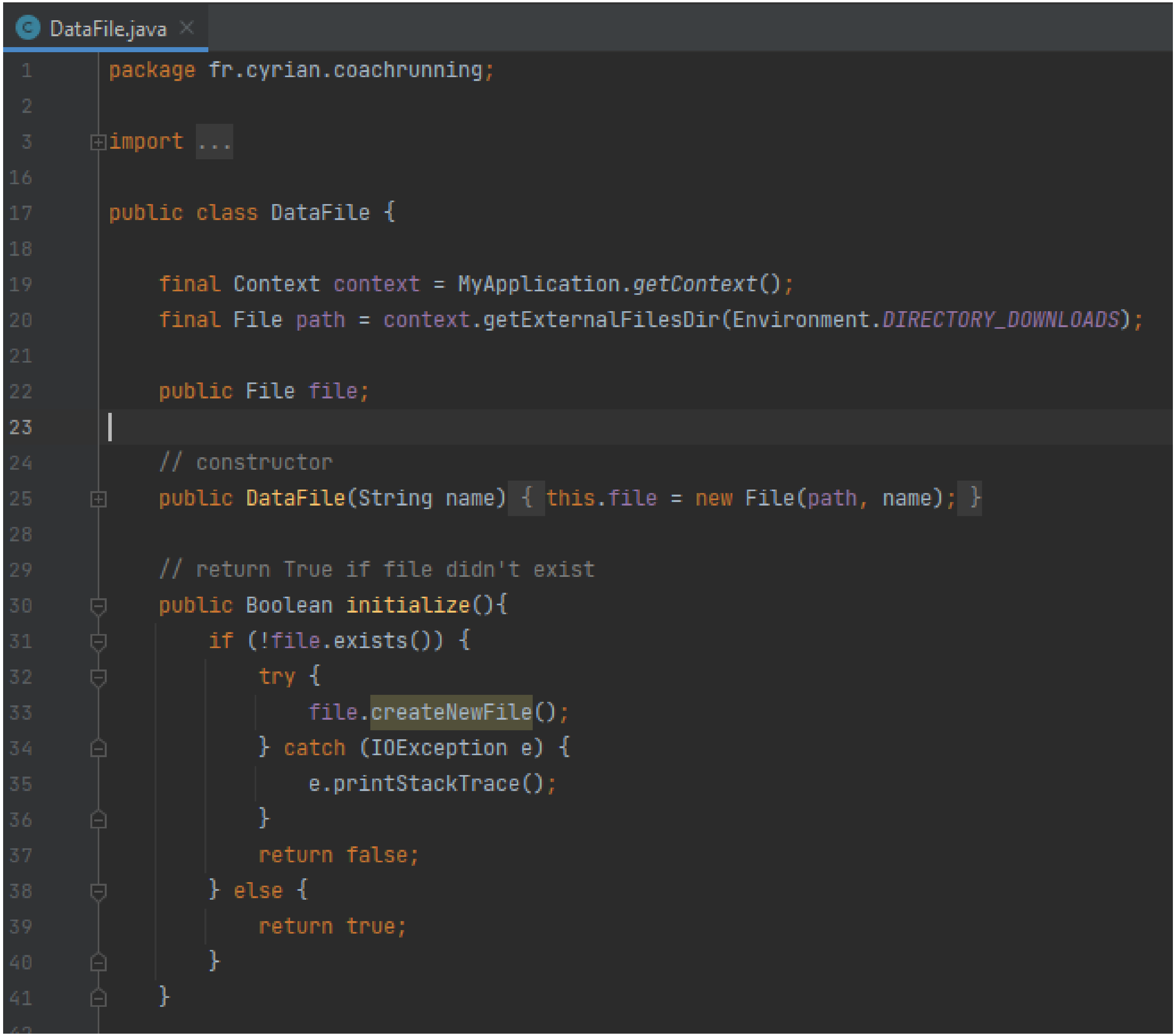
```java
package fr.cyrian.coachrunning;

import ...

public class AppListAdapter extends BaseAdapter {

    private Context context;
    private List<AppListItem> appListItemList;
    private LayoutInflater inflater;
    private MyActionCallback myActionCallback;
    private ArrayList<String[]> newarr;

    // constructor
    public AppListAdapter(Context context, List<AppListItem> appListItemList, MyActionCallback mActionCallback, ArrayList<String[]> newarr) {
        this.context = context;
        this.appListItemList = appListItemList;
        this.inflater = LayoutInflater.from(context);
        this.myActionCallback = mActionCallback;
        this.newarr = newarr;
    }

    @Override
    public int getCount() { return appListItemList.size(); }

    @Override
    public AppListItem getItem(int position) { return appListItemList.get(position); }

    @Override
    public long getItemId(int position) { return 0; }
```



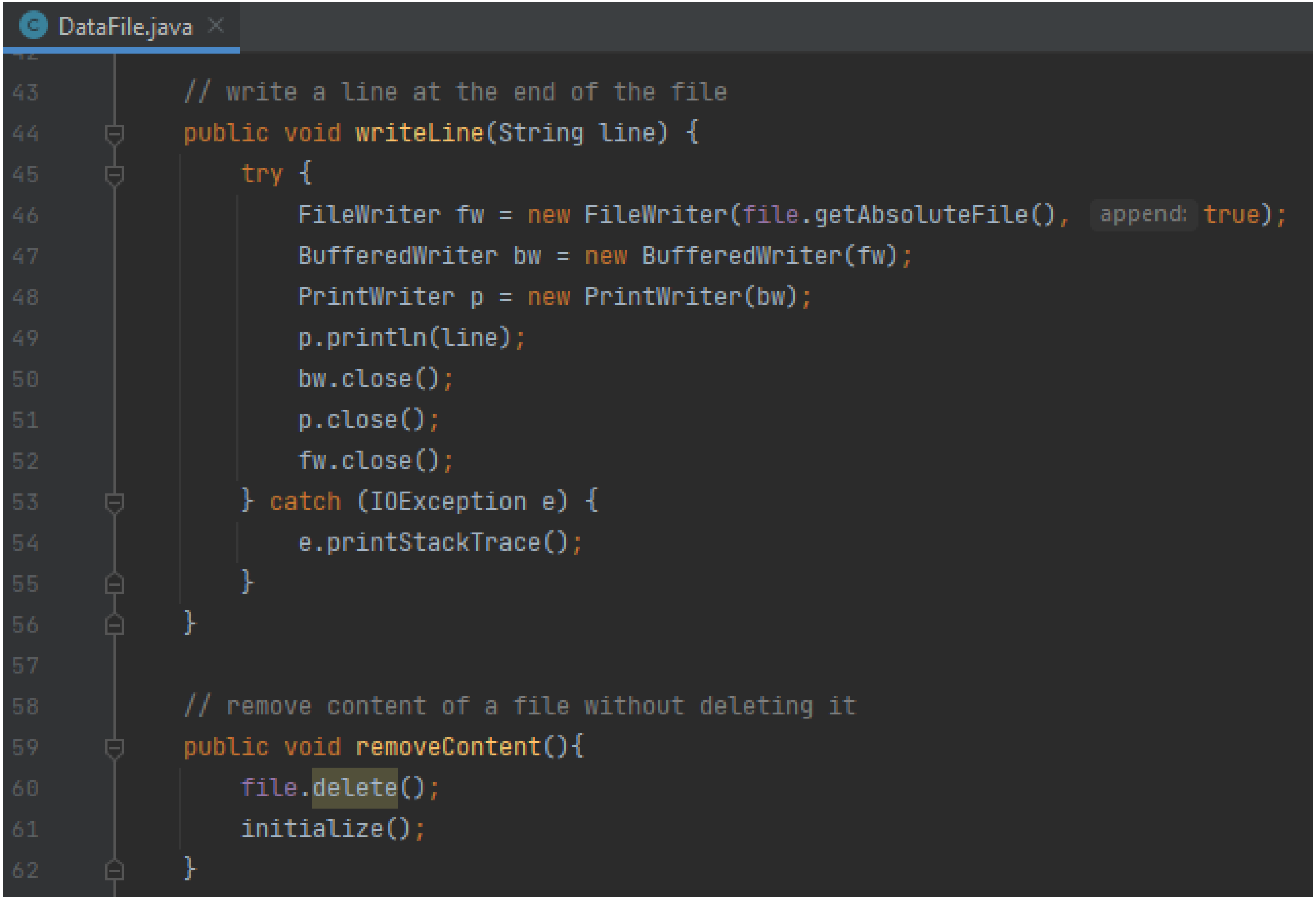
```java
    @Override
    public View getView(int position, View view, ViewGroup parent) {

        view = inflater.inflate(R.layout.app_list_adapter_item,  root: null);

        // get infos about item
        AppListItem currentItem = getItem(position);

        String itemName = currentItem.getName();
        String pkgName = currentItem.getPkgName();
        Boolean isChecked = currentItem.isChecked();
        Drawable itemIcon = currentItem.getImg();

        //change check
        CheckBox checkbox = (CheckBox) view.findViewById(R.id.checkbox);
        checkbox.setChecked(Boolean.valueOf(newarr.get(position)[1]));

        //change item name view
        TextView itemNameView = view.findViewById(R.id.item_name);
        itemNameView.setText(itemName);

        // change item img view
        ImageView itemIconView = view.findViewById(R.id.item_icon);
        itemIconView.setImageDrawable(itemIcon);

```



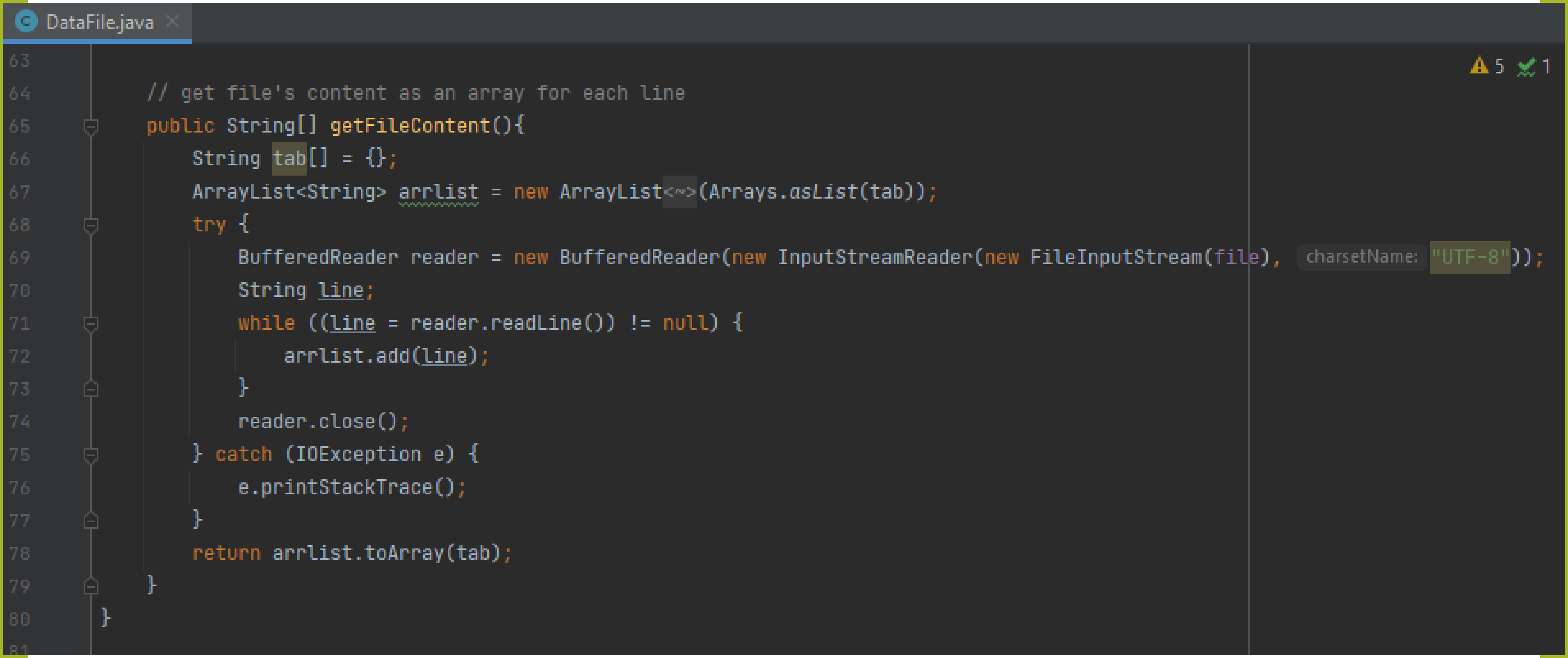
```java
        // click listener
        checkbox.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String pkg = newarr.get(position)[0];
                if(((CompoundButton) v).isChecked()) {
                    //Toast.makeText(context, "checked", Toast.LENGTH_SHORT).show();
                    //checkbox.setChecked(isChecked);
                    newarr.set(position, new String[]{pkg, "true"});
                }else {
                    //Toast.makeText(context, "uncheked", Toast.LENGTH_SHORT).show();
                    //checkbox.setChecked(!isChecked);
                    newarr.set(position, new String[]{pkg, "false"});
                }
                myActionCallback.onCheckboxClick(position,((CompoundButton) v).isChecked());
            }
        });
        return view;
    }
}
```



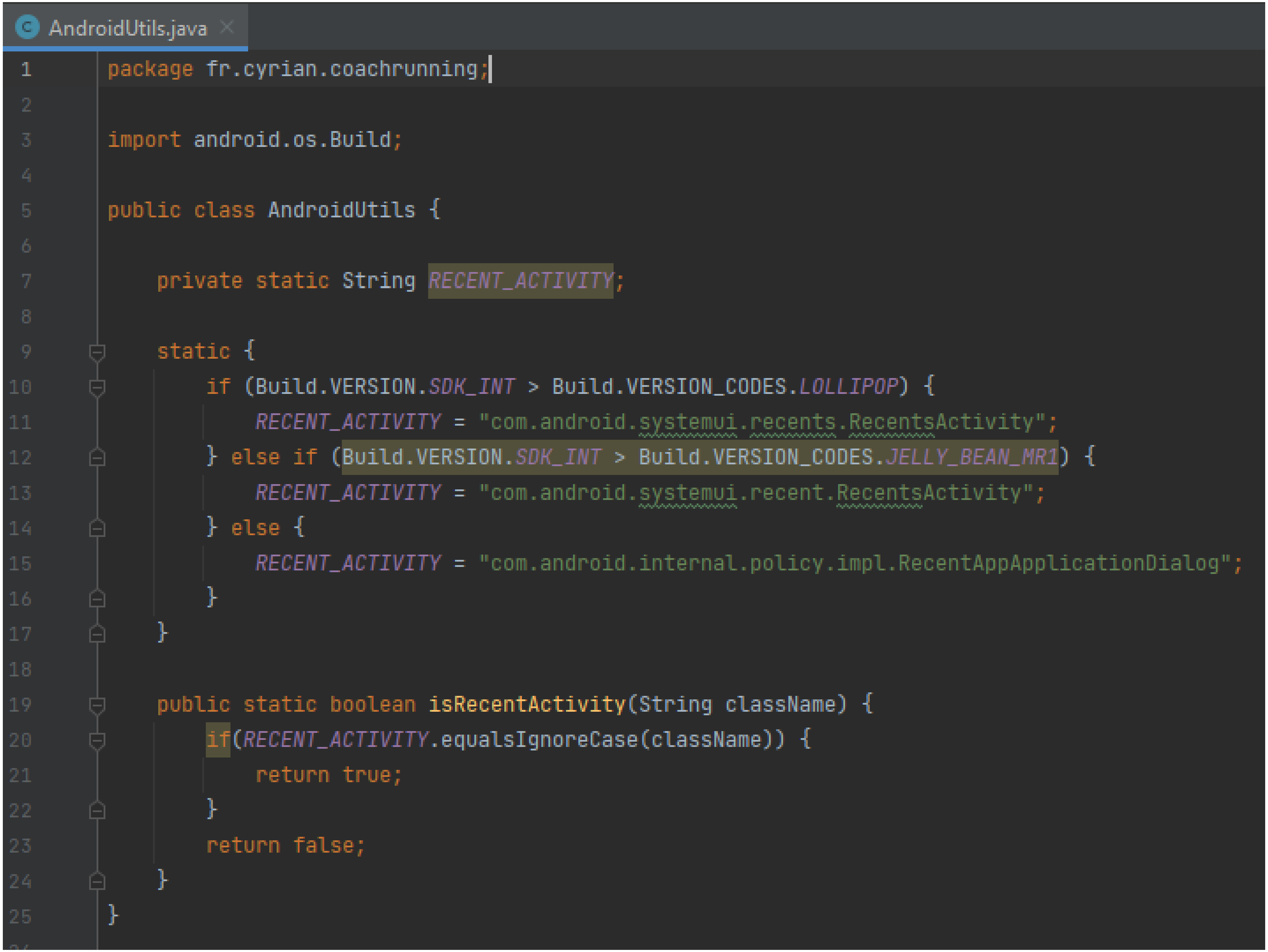
```java
package fr.cyrian.coachrunning;

import ...

public class DataFile {

    final Context context = MyApplication.getContext();
    final File path = context.getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS);

    public File file;

    // constructor
    public DataFile(String name) { this.file = new File(path, name); }

    // return True if file didn't exist
    public Boolean initialize(){
        if (!file.exists()) {
            try {
                file.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return false;
        } else {
            return true;
        }
    }
```



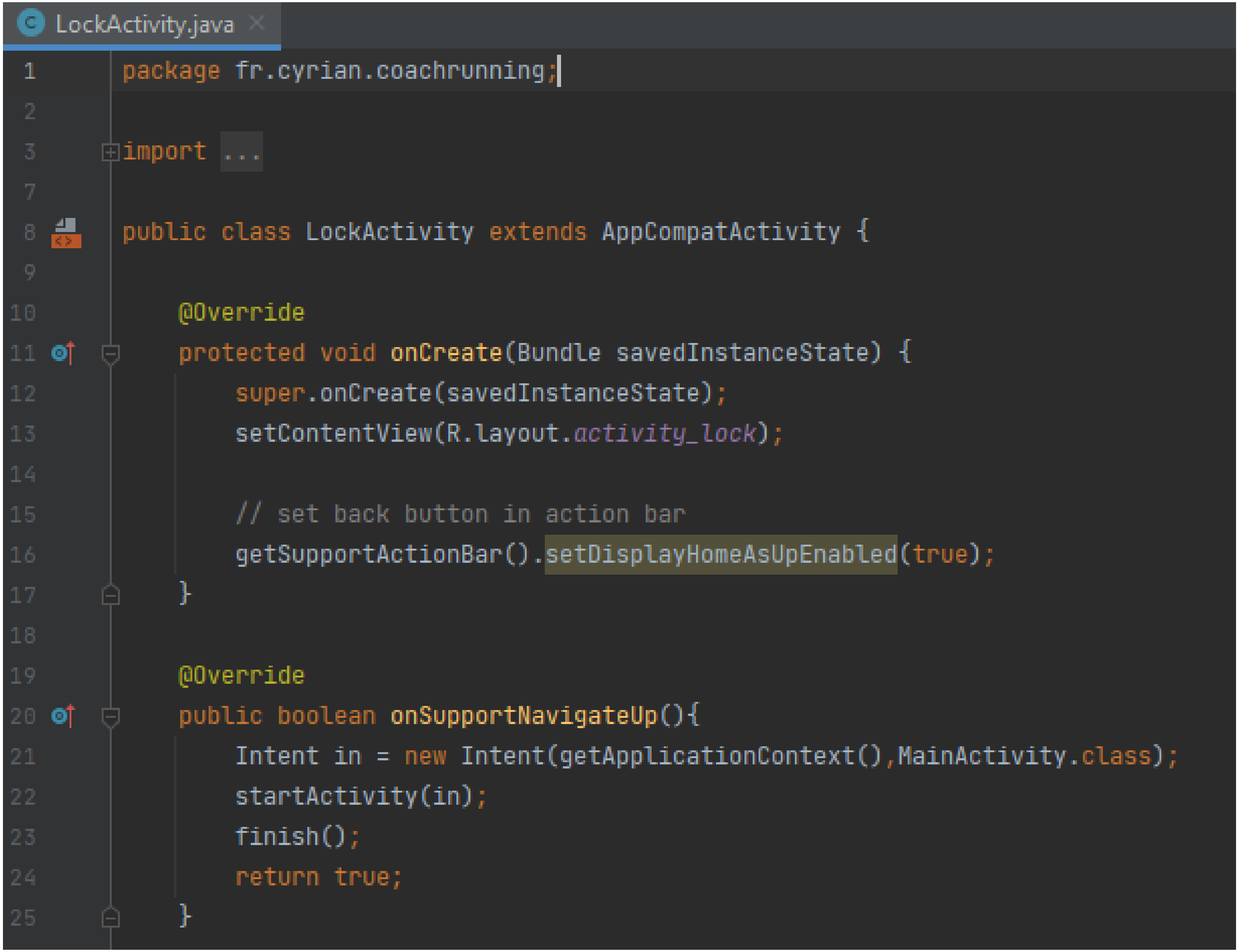
```java
    // write a line at the end of the file
    public void writeLine(String line) {
        try {
            FileWriter fw = new FileWriter(file.getAbsoluteFile(),  append: true);
            BufferedWriter bw = new BufferedWriter(fw);
            PrintWriter p = new PrintWriter(bw);
            p.println(line);
            bw.close();
            p.close();
            fw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // remove content of a file without deleting it
    public void removeContent(){
        file.delete();
        initialize();
    }
```



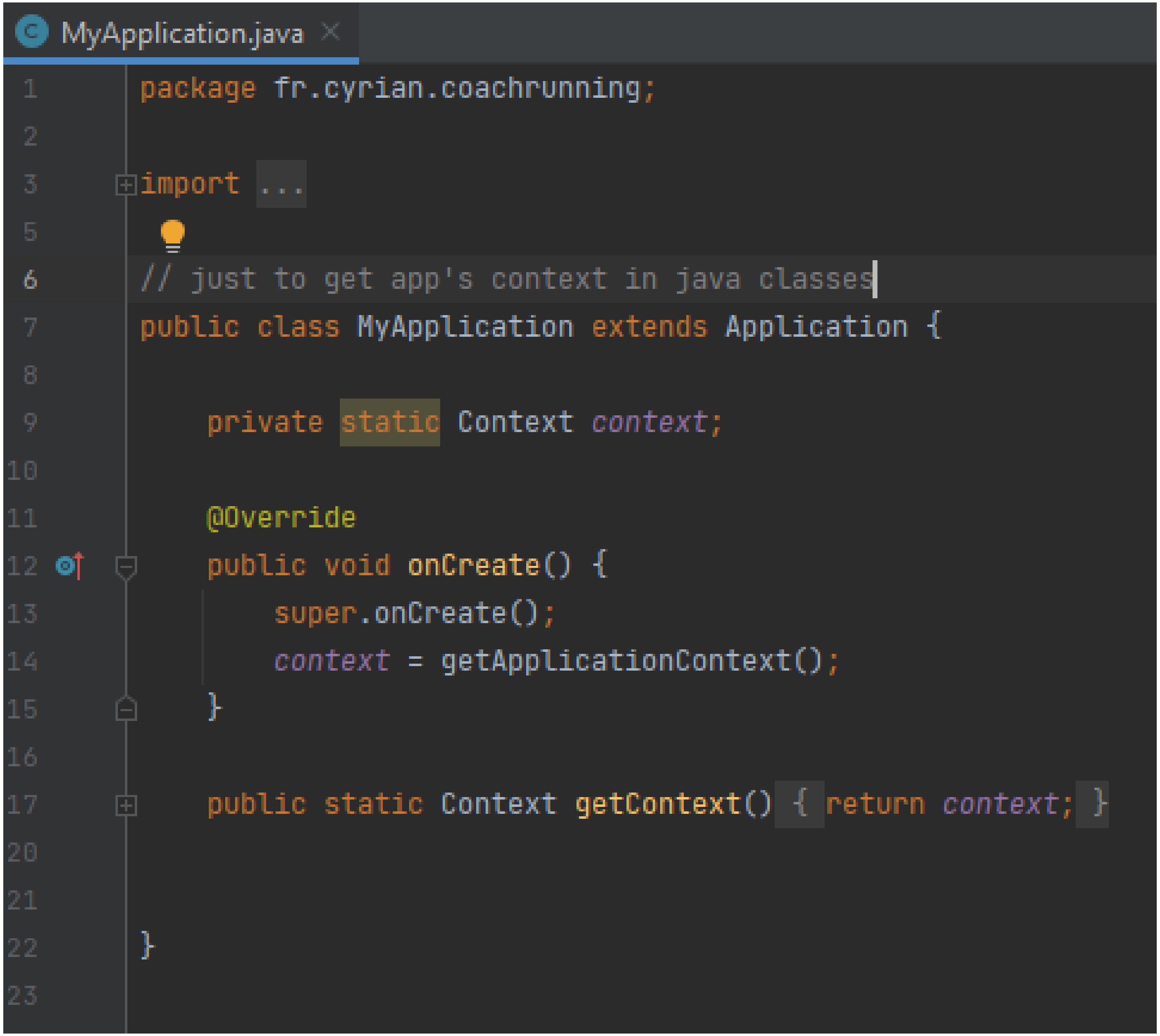
```java

    // get file's content as an array for each line
    public String[] getFileContent(){
        String tab[] = {};
        ArrayList<String> arrlist = new ArrayList<~>(Arrays.asList(tab));
        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(new FileInputStream(file), charsetName: "UTF-8"));
            String line;
            while ((line = reader.readLine()) != null) {
                arrlist.add(line);
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return arrlist.toArray(tab);
    }
}
```



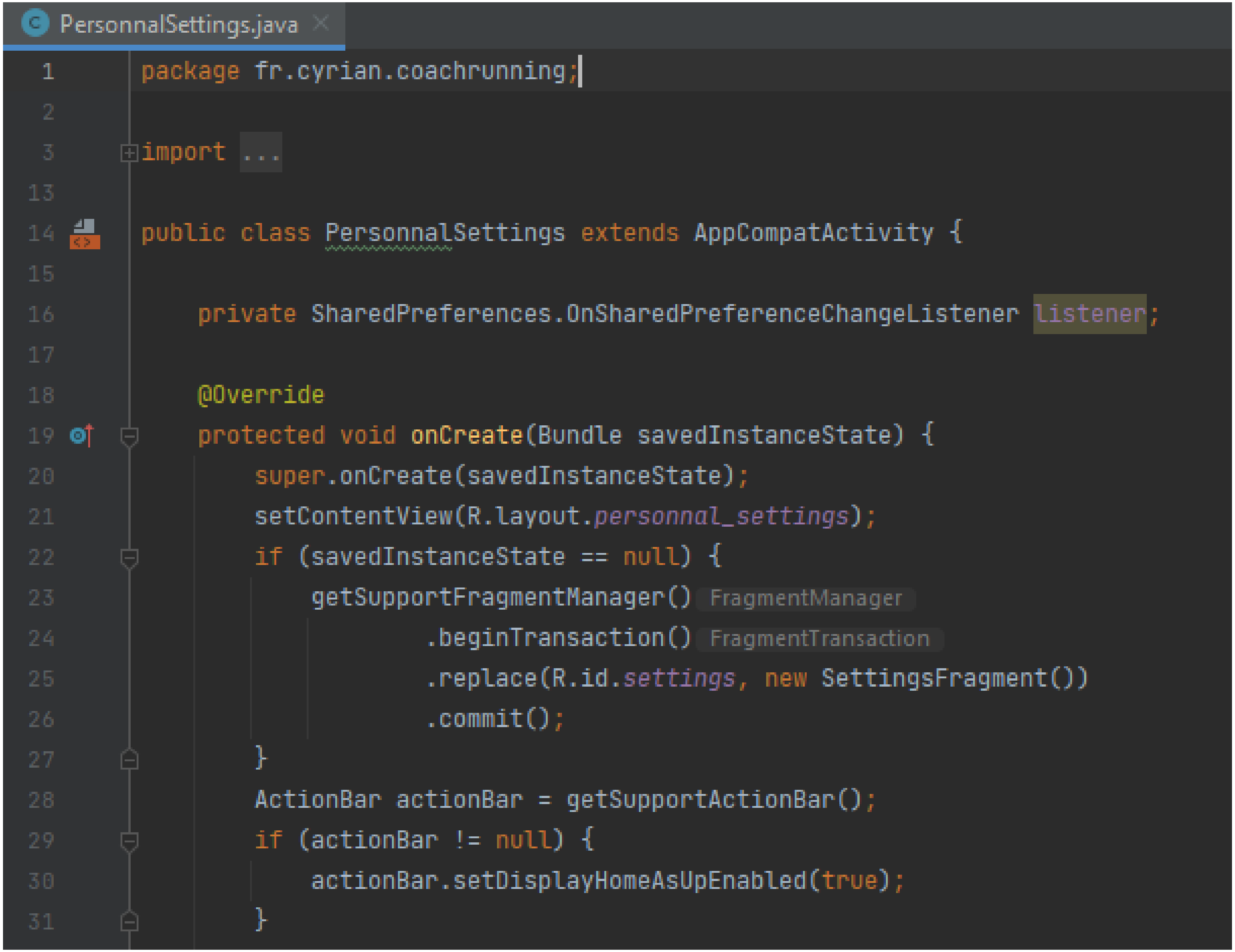
```java
package fr.cyrian.coachrunning;

import android.os.Build;

public class AndroidUtils {

    private static String RECENT_ACTIVITY;

    static {
        if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP) {
            RECENT_ACTIVITY = "com.android.systemui.recents.RecentsActivity";
        } else if (Build.VERSION.SDK_INT > Build.VERSION_CODES.JELLY_BEAN_MR1) {
            RECENT_ACTIVITY = "com.android.systemui.recent.RecentsActivity";
        } else {
            RECENT_ACTIVITY = "com.android.internal.policy.impl.RecentAppApplicationDialog";
        }
    }

    public static boolean isRecentActivity(String className) {
        if(RECENT_ACTIVITY.equalsIgnoreCase(className)) {
            return true;
        }
        return false;
    }
}
```



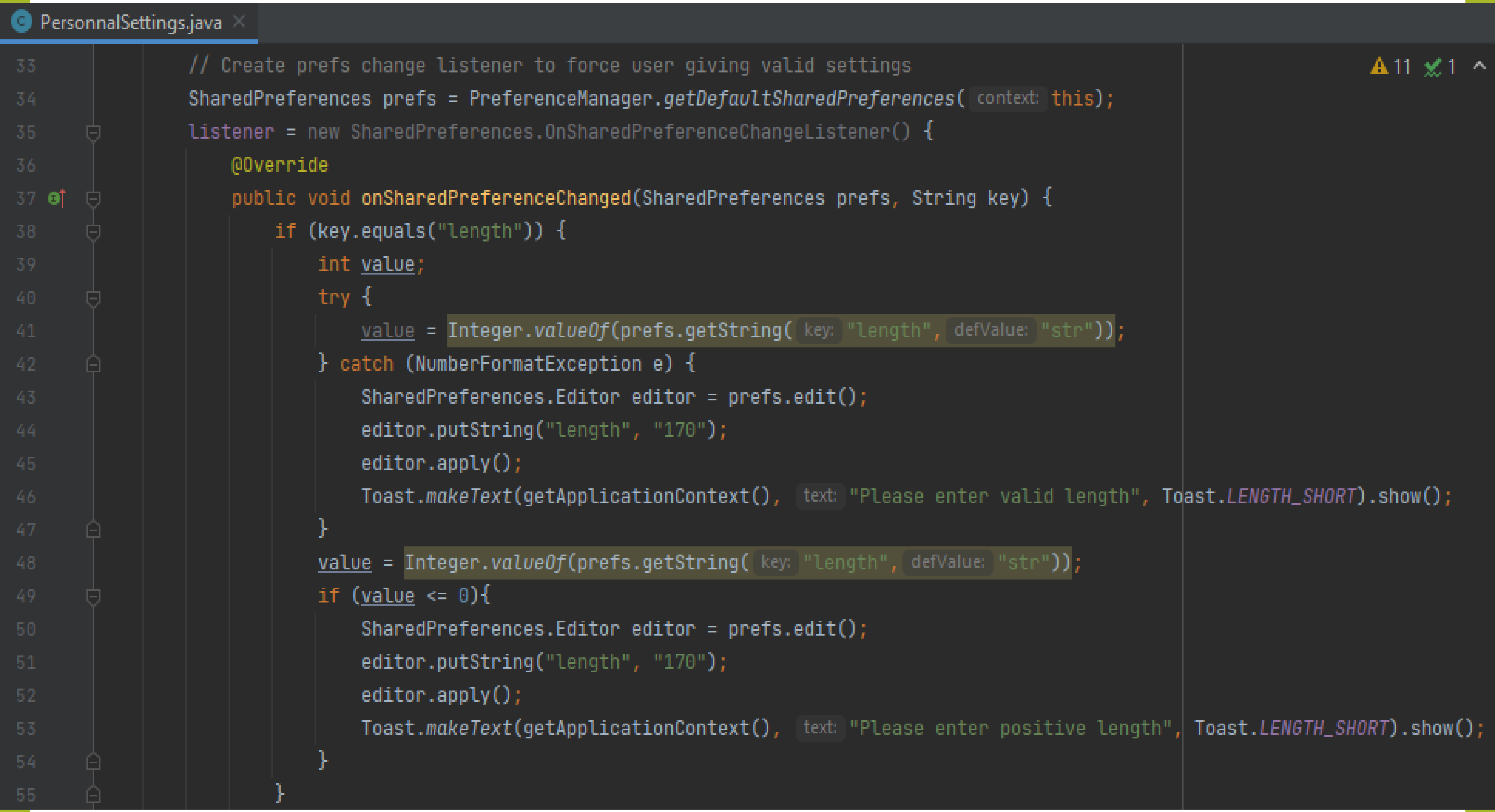
```java
package fr.cyrian.coachrunning;

import ...

public class LockActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lock);

        // set back button in action bar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onSupportNavigateUp(){
        Intent in = new Intent(getApplicationContext(),MainActivity.class);
        startActivity(in);
        finish();
        return true;
    }
```



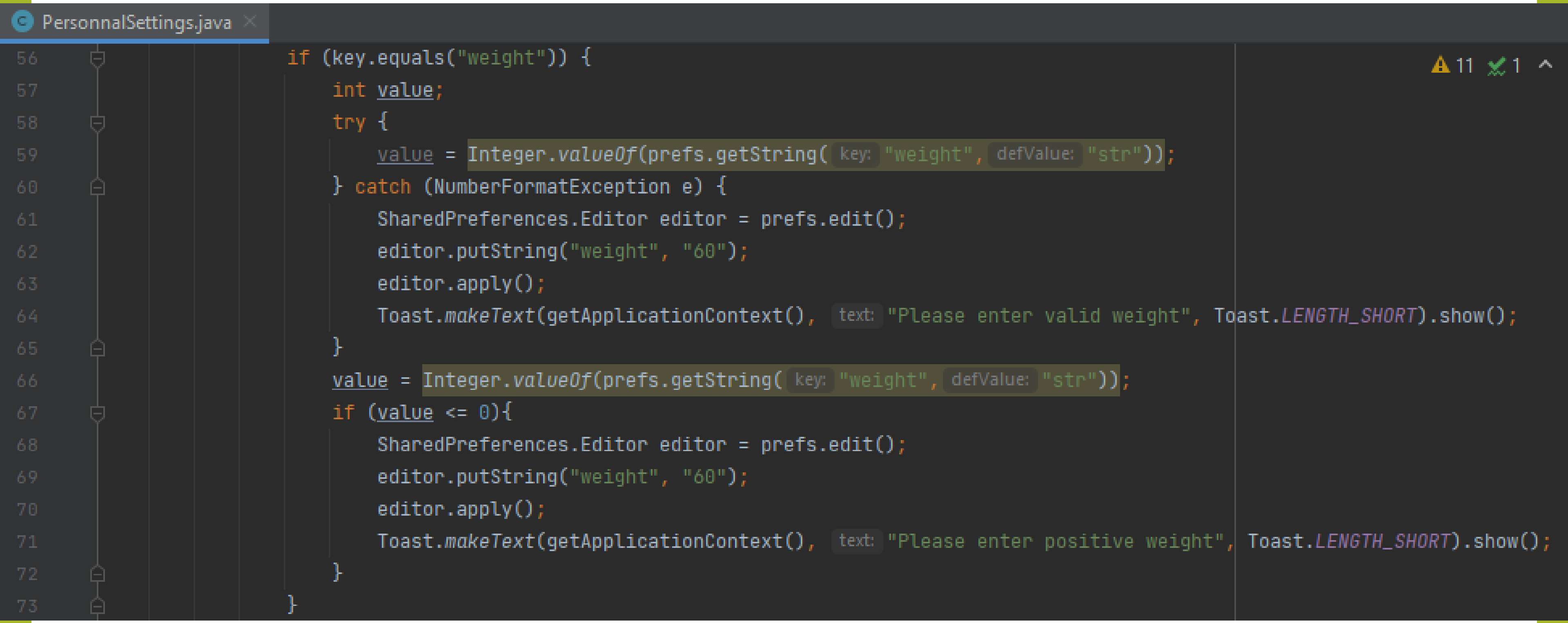
```java
package fr.cyrian.coachrunning;

import ...

// just to get app's context in java classes
public class MyApplication extends Application {

    private static Context context;

    @Override
    public void onCreate() {
        super.onCreate();
        context = getApplicationContext();
    }

    public static Context getContext() { return context; }


}

```



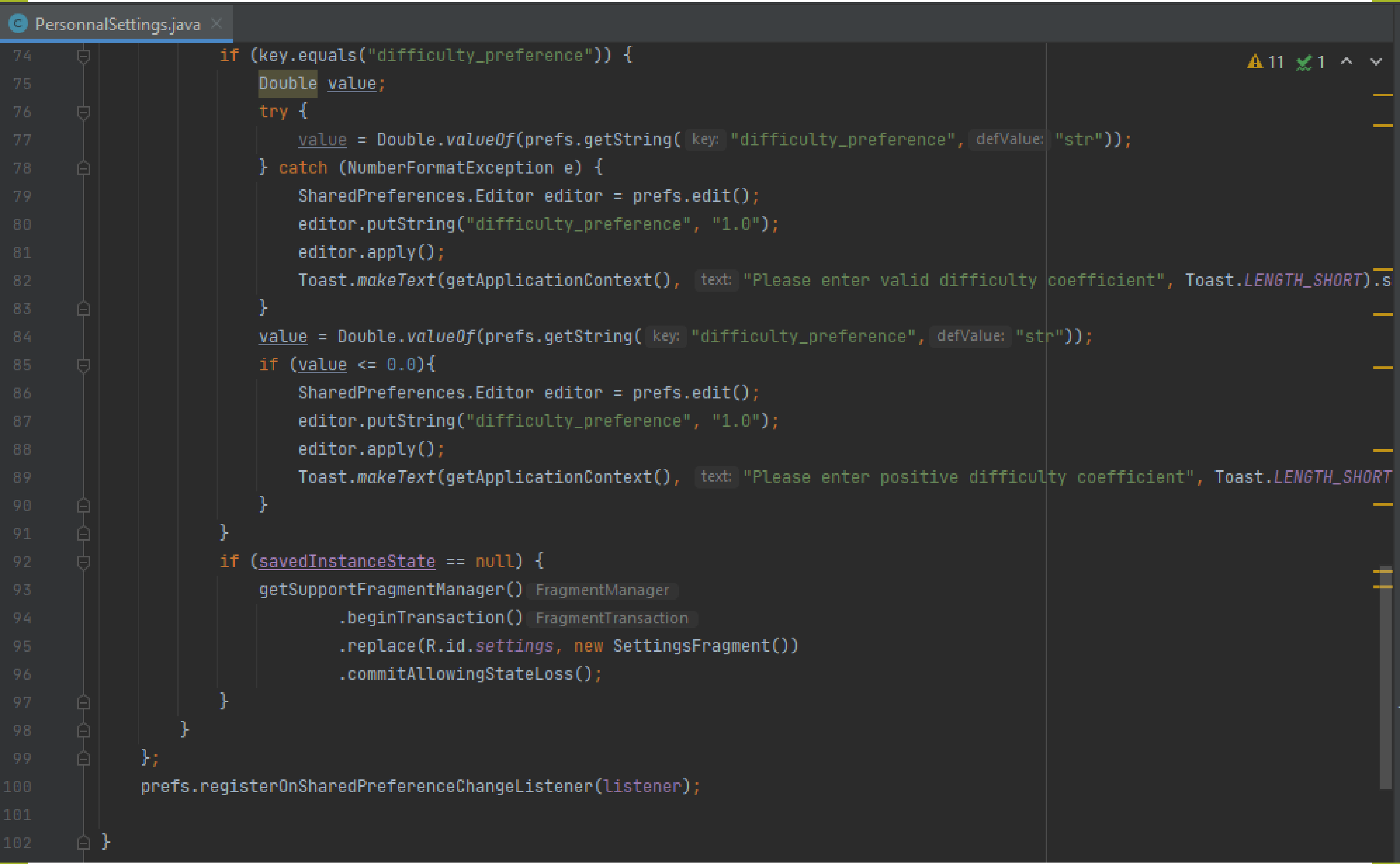
```java
package fr.cyrian.coachrunning;

import ...

public class PersonnalSettings extends AppCompatActivity {

    private SharedPreferences.OnSharedPreferenceChangeListener listener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.personnal_settings);
        if (savedInstanceState == null) {
            getSupportFragmentManager()
                    .beginTransaction()
                    .replace(R.id.settings, new SettingsFragment())
                    .commit();
        }
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.setDisplayHomeAsUpEnabled(true);
        }
```



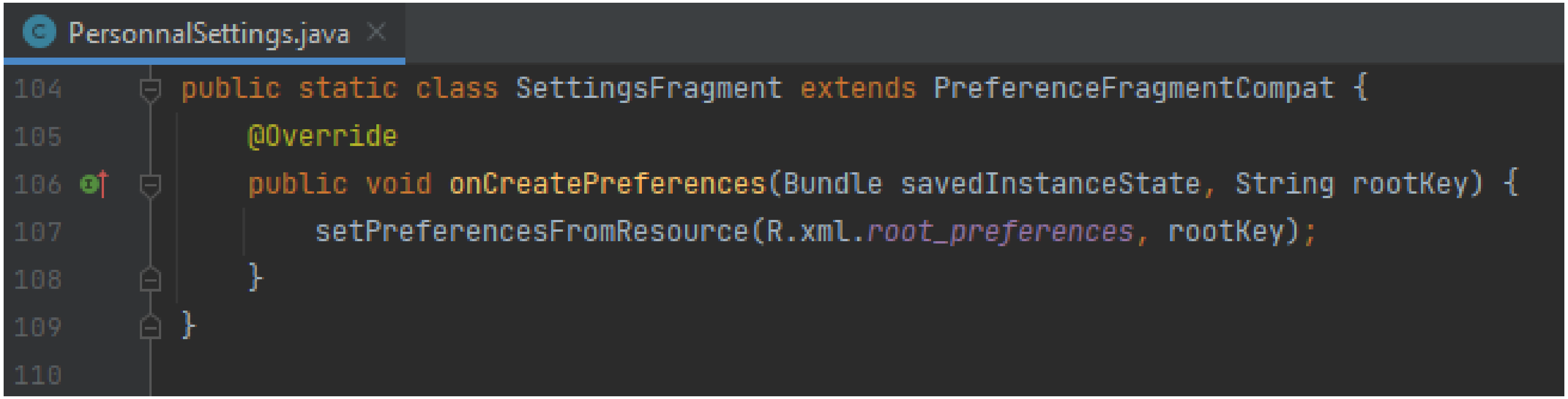
```java
        // Create prefs change listener to force user giving valid settings
        SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences( context: this);
        listener = new SharedPreferences.OnSharedPreferenceChangeListener() {
            @Override
            public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {
                if (key.equals("length")) {
                    int value;
                    try {
                        value = Integer.valueOf(prefs.getString( key: "length", defValue: "str"));
                    } catch (NumberFormatException e) {
                        SharedPreferences.Editor editor = prefs.edit();
                        editor.putString("length", "170");
                        editor.apply();
                        Toast.makeText(getApplicationContext(),  text: "Please enter valid length", Toast.LENGTH_SHORT).show();
                    }
                    value = Integer.valueOf(prefs.getString( key: "length", defValue: "str"));
                    if (value <= 0){
                        SharedPreferences.Editor editor = prefs.edit();
                        editor.putString("length", "170");
                        editor.apply();
                        Toast.makeText(getApplicationContext(),  text: "Please enter positive length", Toast.LENGTH_SHORT).show();
                    }
                }
```



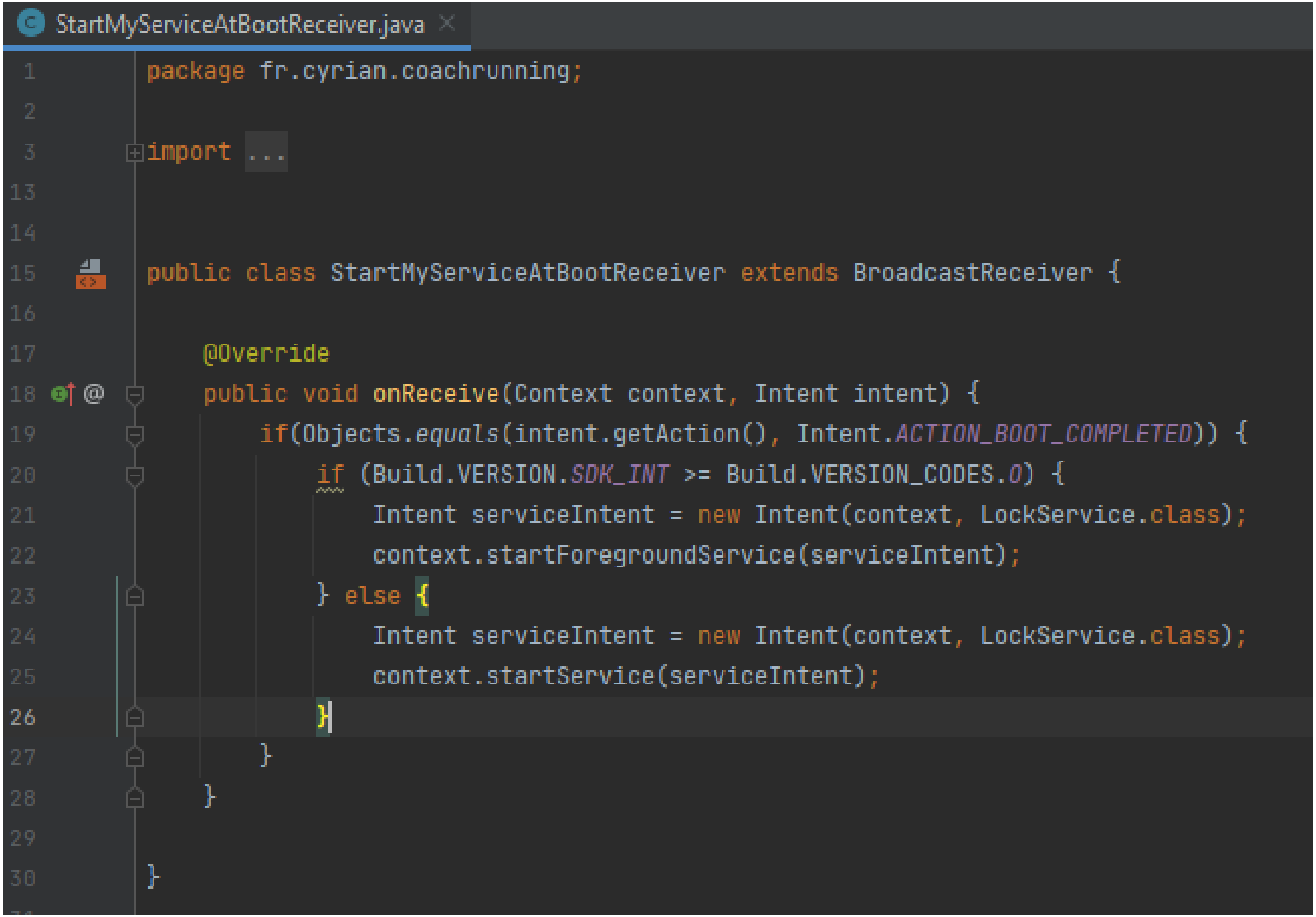
```
if (key.equals("weight")) {
    int value;
    try {
        value = Integer.valueOf(prefs.getString( key: "weight", defValue: "str"));
    } catch (NumberFormatException e) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("weight", "60");
        editor.apply();
        Toast.makeText(getApplicationContext(), text: "Please enter valid weight", Toast.LENGTH_SHORT).show();
    }
    value = Integer.valueOf(prefs.getString( key: "weight", defValue: "str"));
    if (value <= 0){
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("weight", "60");
        editor.apply();
        Toast.makeText(getApplicationContext(), text: "Please enter positive weight", Toast.LENGTH_SHORT).show();
    }
}
```



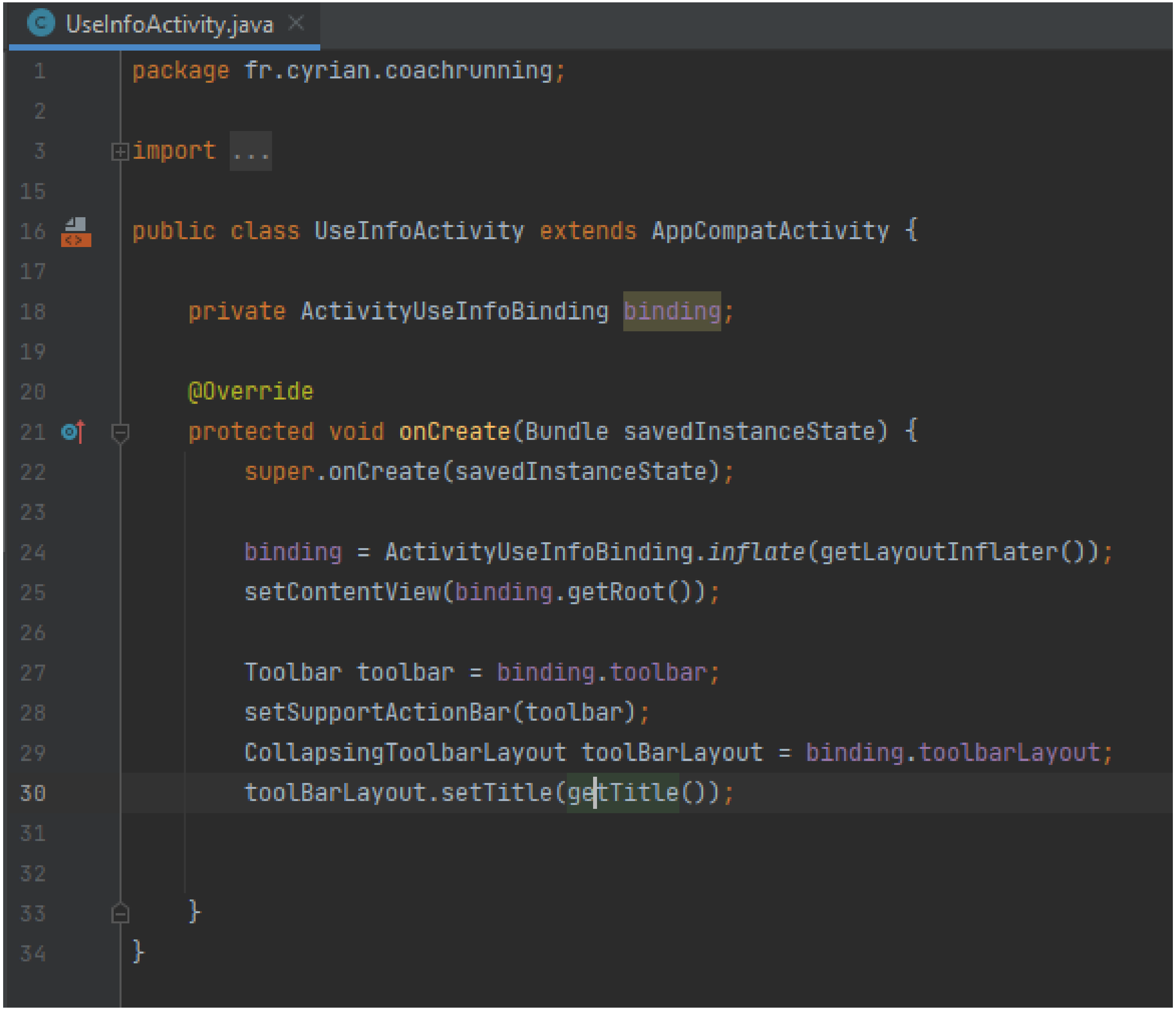
```
            if (key.equals("difficulty_preference")) {
                Double value;
                try {
                    value = Double.valueOf(prefs.getString( key: "difficulty_preference", defValue: "str"));
                } catch (NumberFormatException e) {
                    SharedPreferences.Editor editor = prefs.edit();
                    editor.putString("difficulty_preference", "1.0");
                    editor.apply();
                    Toast.makeText(getApplicationContext(), text: "Please enter valid difficulty coefficient", Toast.LENGTH_SHORT).s
                }
                value = Double.valueOf(prefs.getString( key: "difficulty_preference", defValue: "str"));
                if (value <= 0.0){
                    SharedPreferences.Editor editor = prefs.edit();
                    editor.putString("difficulty_preference", "1.0");
                    editor.apply();
                    Toast.makeText(getApplicationContext(), text: "Please enter positive difficulty coefficient", Toast.LENGTH_SHORT
                }
            }
            if (savedInstanceState == null) {
                getSupportFragmentManager() FragmentManager
                        .beginTransaction() FragmentTransaction
                        .replace(R.id.settings, new SettingsFragment())
                        .commitAllowingStateLoss();
            }
        }
    };
    prefs.registerOnSharedPreferenceChangeListener(listener);

}
```



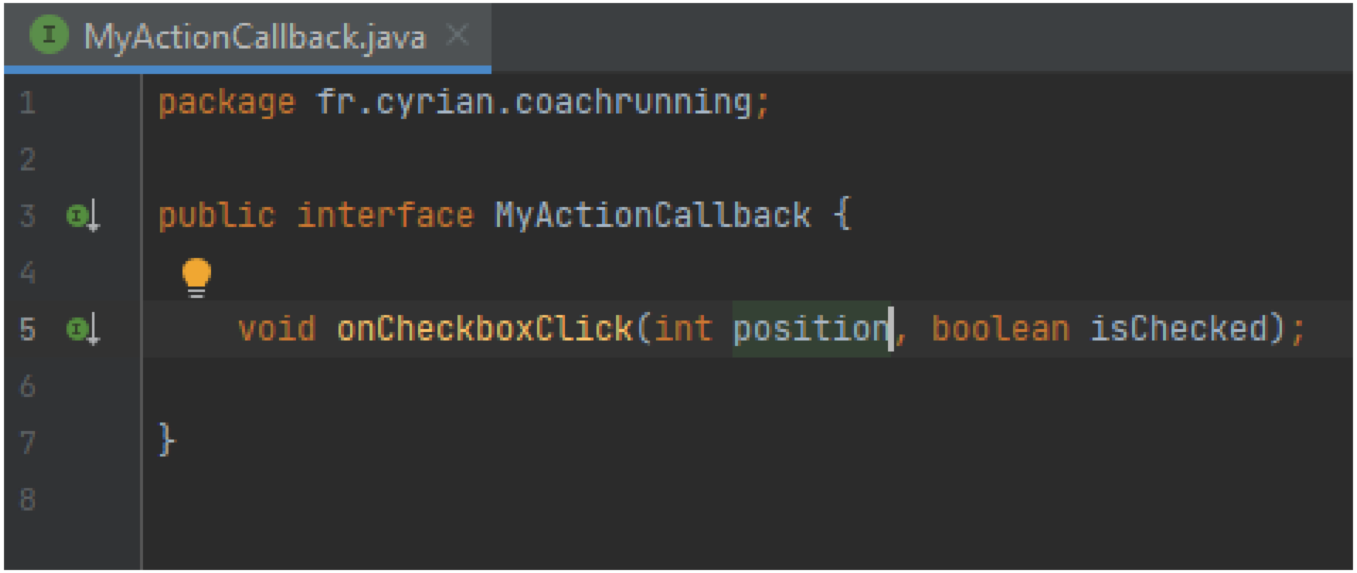
```
public static class SettingsFragment extends PreferenceFragmentCompat {
    @Override
    public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
        setPreferencesFromResource(R.xml.root_preferences, rootKey);
    }
}
```



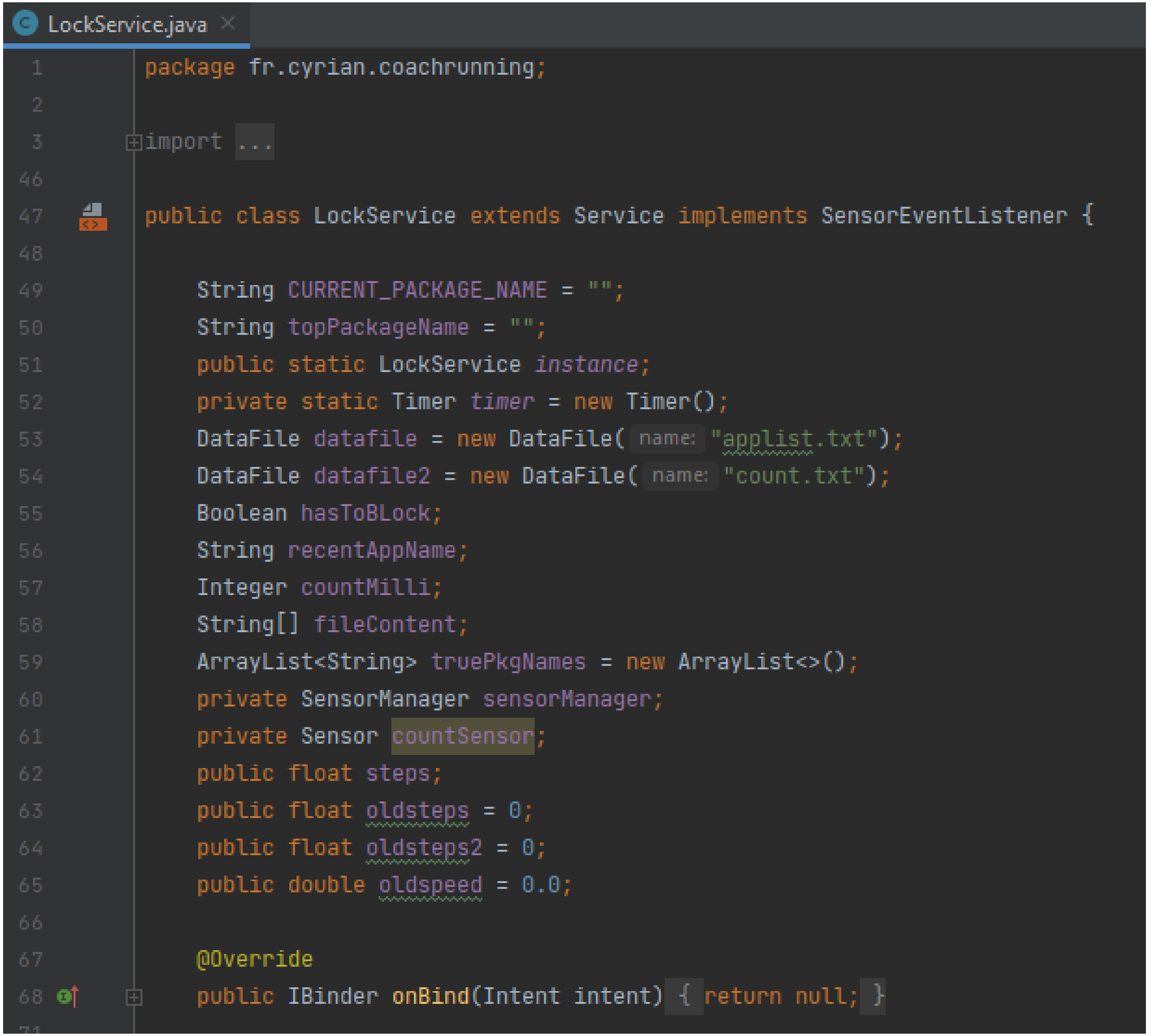
```java
package fr.cyrian.coachrunning;

import ...


public class StartMyServiceAtBootReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if(Objects.equals(intent.getAction(), Intent.ACTION_BOOT_COMPLETED)) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                Intent serviceIntent = new Intent(context, LockService.class);
                context.startForegroundService(serviceIntent);
            } else {
                Intent serviceIntent = new Intent(context, LockService.class);
                context.startService(serviceIntent);
            }
        }
    }

}
```



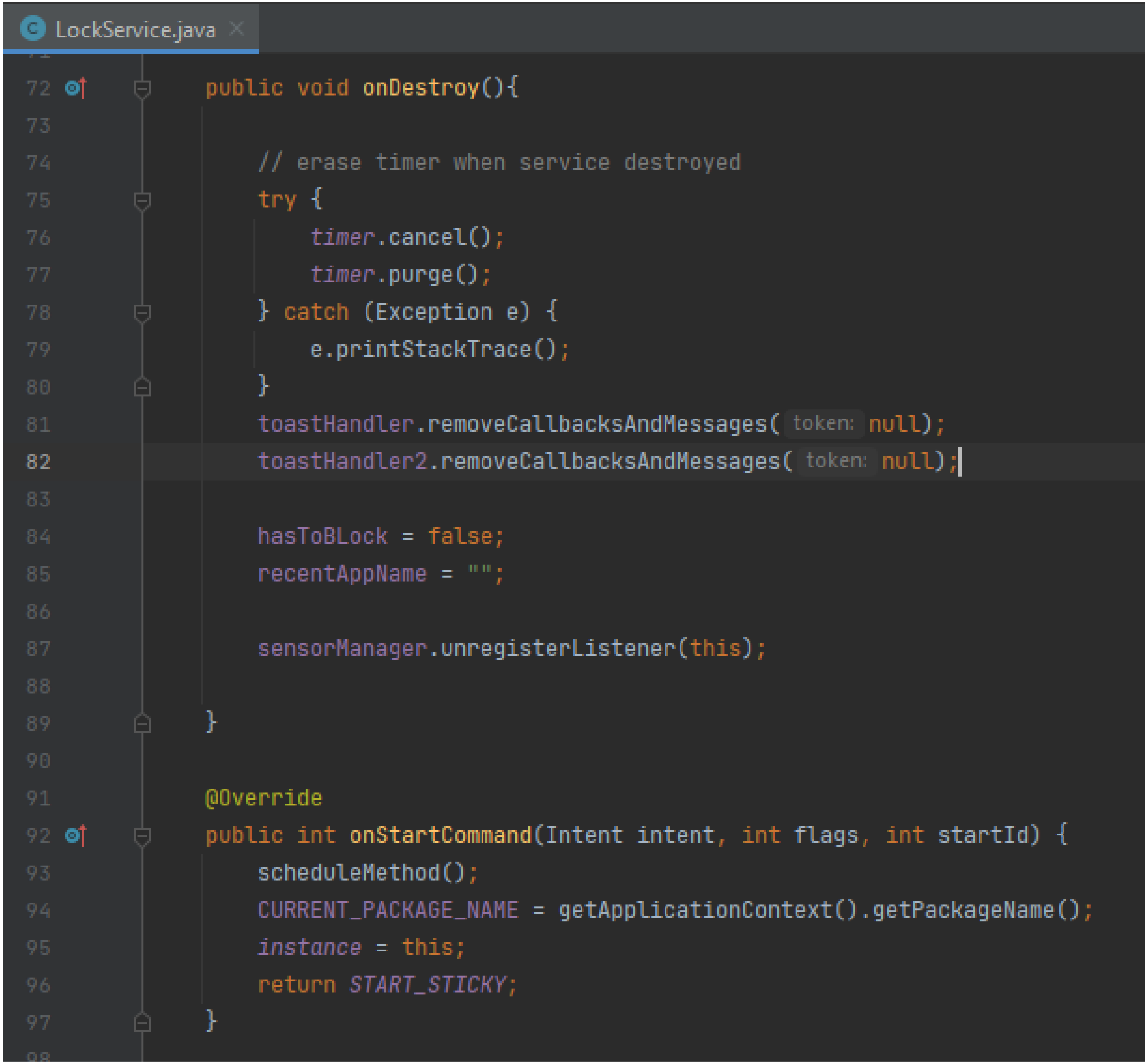
```java
package fr.cyrian.coachrunning;

import ...

public class UseInfoActivity extends AppCompatActivity {

    private ActivityUseInfoBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityUseInfoBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        Toolbar toolbar = binding.toolbar;
        setSupportActionBar(toolbar);
        CollapsingToolbarLayout toolBarLayout = binding.toolbarLayout;
        toolBarLayout.setTitle(getTitle());


    }
}
```



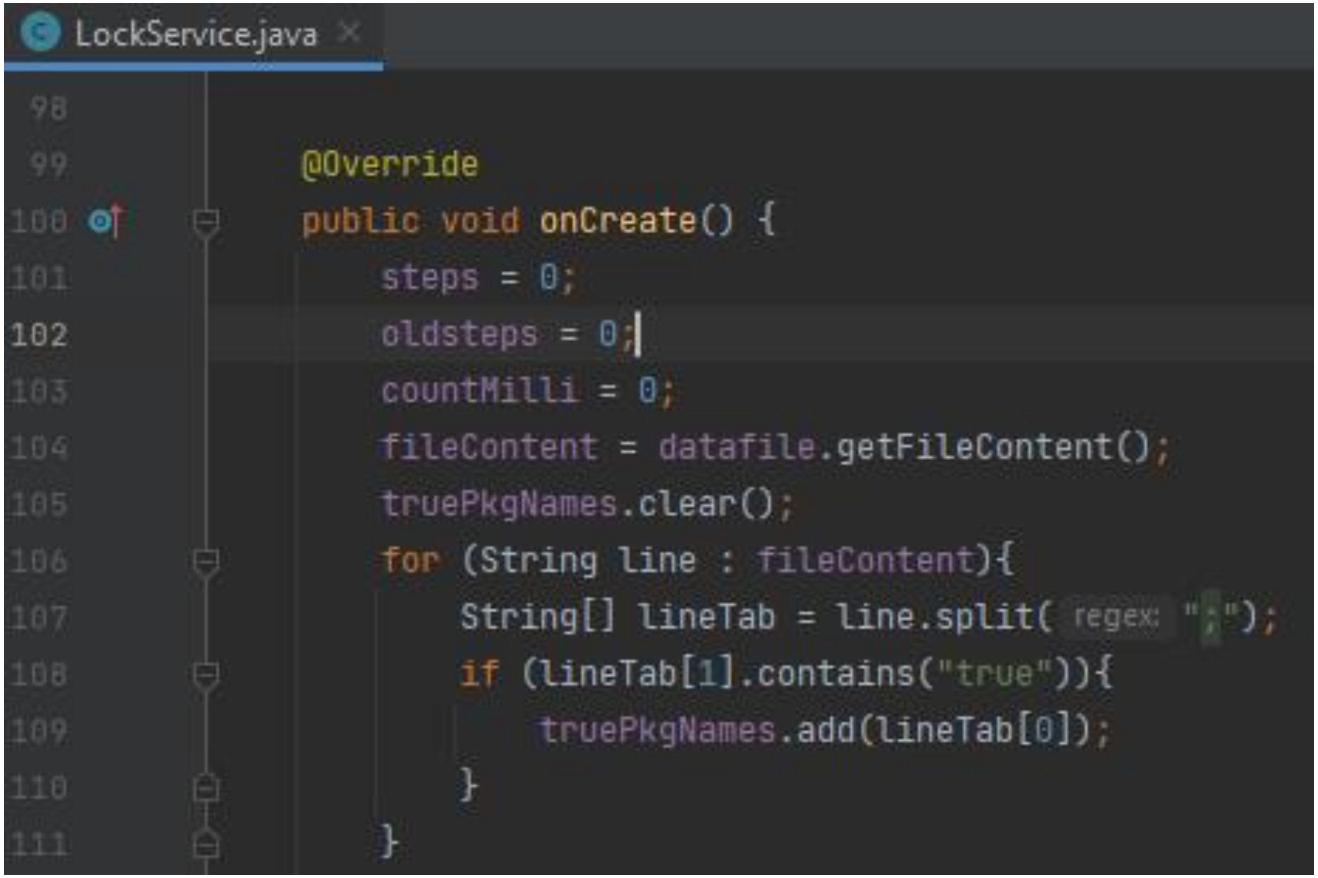
```java
package fr.cyrian.coachrunning;

public interface MyActionCallback {

    void onCheckboxClick(int position, boolean isChecked);

}

```



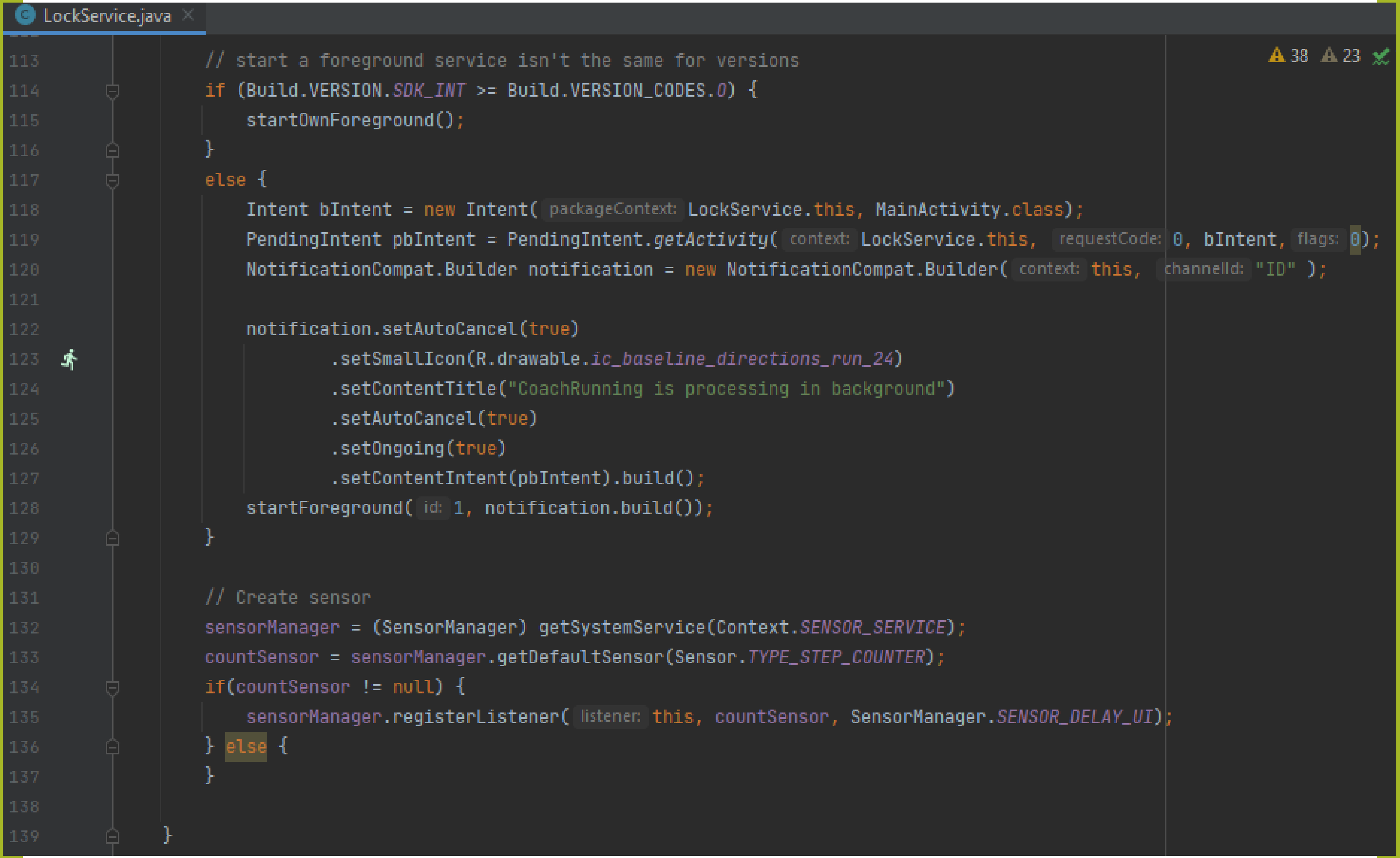
```java
package fr.cyrian.coachrunning;

import ...

public class LockService extends Service implements SensorEventListener {

    String CURRENT_PACKAGE_NAME = "";
    String topPackageName = "";
    public static LockService instance;
    private static Timer timer = new Timer();
    DataFile datafile = new DataFile( name: "applist.txt");
    DataFile datafile2 = new DataFile( name: "count.txt");
    Boolean hasToBLock;
    String recentAppName;
    Integer countMilli;
    String[] fileContent;
    ArrayList<String> truePkgNames = new ArrayList<>();
    private SensorManager sensorManager;
    private Sensor countSensor;
    public float steps;
    public float oldsteps = 0;
    public float oldsteps2 = 0;
    public double oldspeed = 0.0;

    @Override
    public IBinder onBind(Intent intent) { return null; }
```



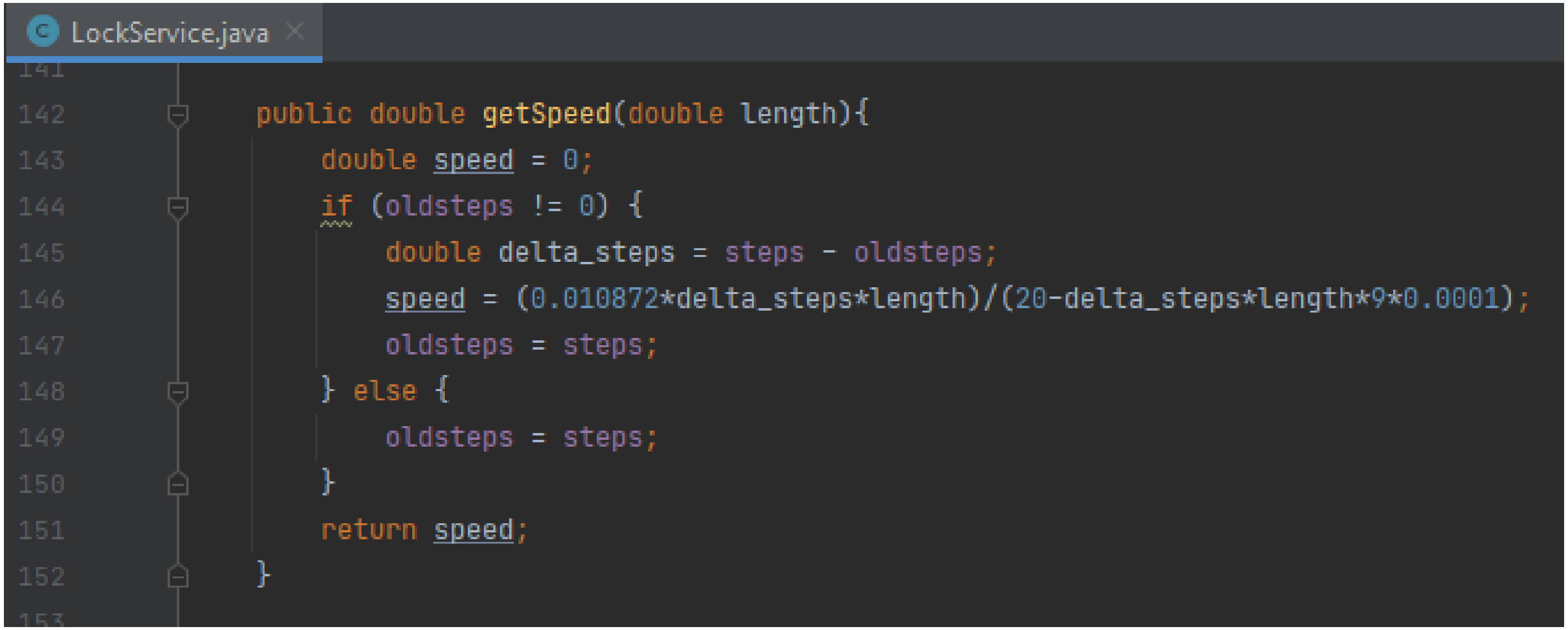
```java
    public void onDestroy(){

        // erase timer when service destroyed
        try {
            timer.cancel();
            timer.purge();
        } catch (Exception e) {
            e.printStackTrace();
        }
        toastHandler.removeCallbacksAndMessages( token: null);
        toastHandler2.removeCallbacksAndMessages( token: null);

        hasToBLock = false;
        recentAppName = "";

        sensorManager.unregisterListener(this);

    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        scheduleMethod();
        CURRENT_PACKAGE_NAME = getApplicationContext().getPackageName();
        instance = this;
        return START_STICKY;
    }
```



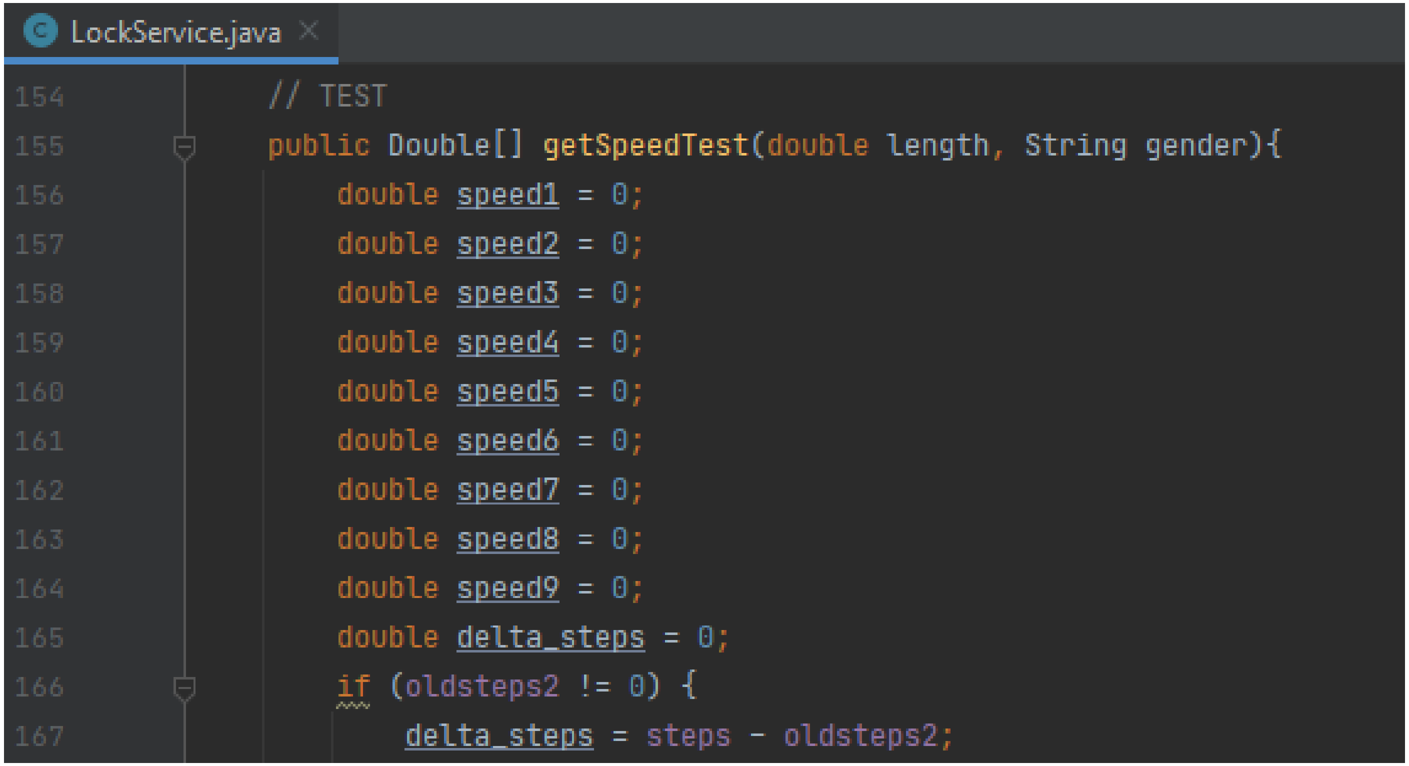
```java

    @Override
    public void onCreate() {
        steps = 0;
        oldsteps = 0;
        countMilli = 0;
        fileContent = datafile.getFileContent();
        truePkgNames.clear();
        for (String line : fileContent){
            String[] lineTab = line.split( regex: ";");
            if (lineTab[1].contains("true")){
                truePkgNames.add(lineTab[0]);
            }
        }
```



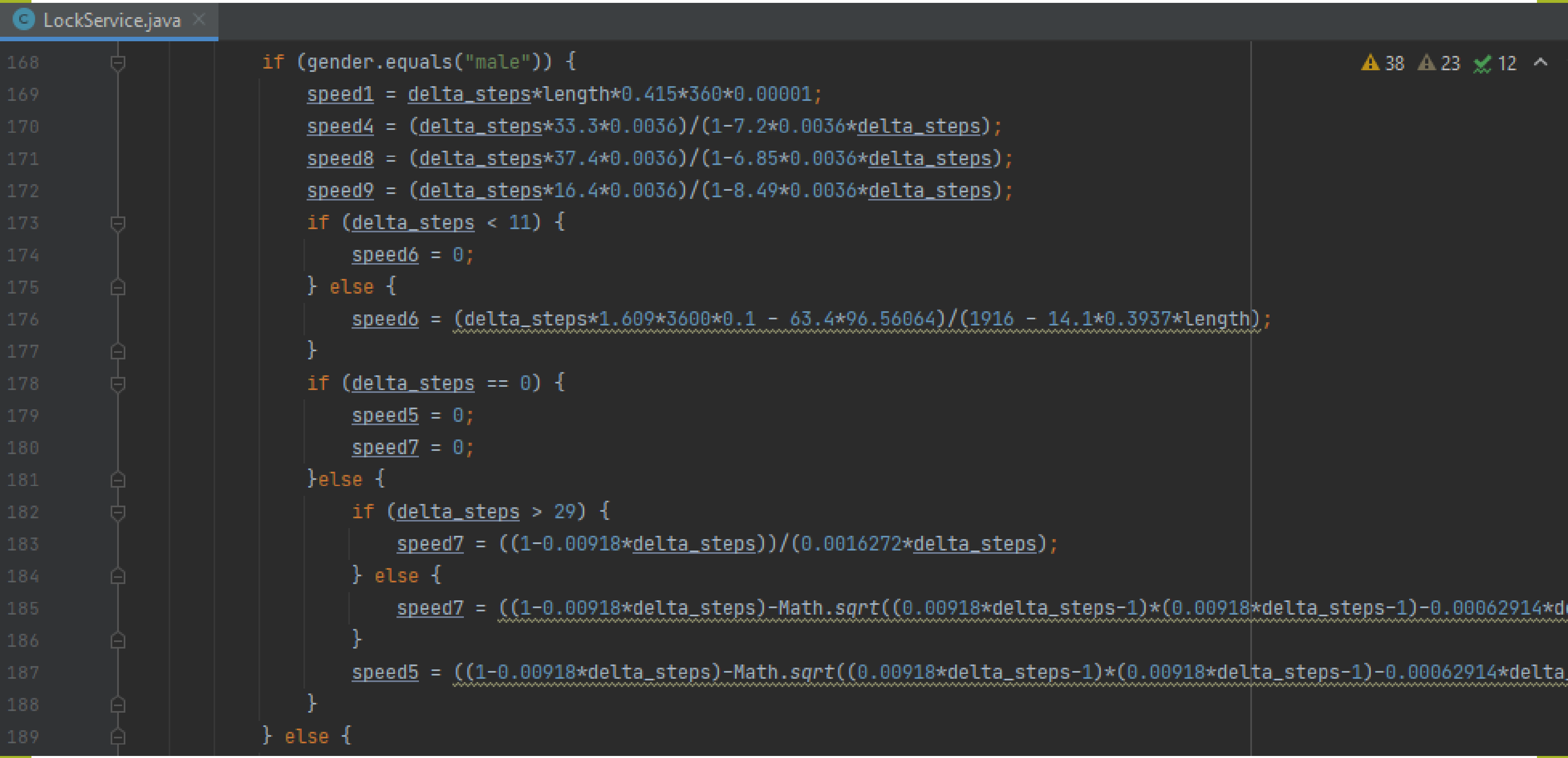
```
        // start a foreground service isn't the same for versions
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            startOwnForeground();
        }
        else {
            Intent bIntent = new Intent( packageContext: LockService.this, MainActivity.class);
            PendingIntent pbIntent = PendingIntent.getActivity( context: LockService.this,  requestCode: 0, bIntent, flags: 0);
            NotificationCompat.Builder notification = new NotificationCompat.Builder( context: this,  channelId: "ID" );

            notification.setAutoCancel(true)
                    .setSmallIcon(R.drawable.ic_baseline_directions_run_24)
                    .setContentTitle("CoachRunning is processing in background")
                    .setAutoCancel(true)
                    .setOngoing(true)
                    .setContentIntent(pbIntent).build();
            startForeground( id: 1, notification.build());
        }

        // Create sensor
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        countSensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);
        if(countSensor != null) {
            sensorManager.registerListener( listener: this, countSensor, SensorManager.SENSOR_DELAY_UI);
        } else {
        }

    }
```



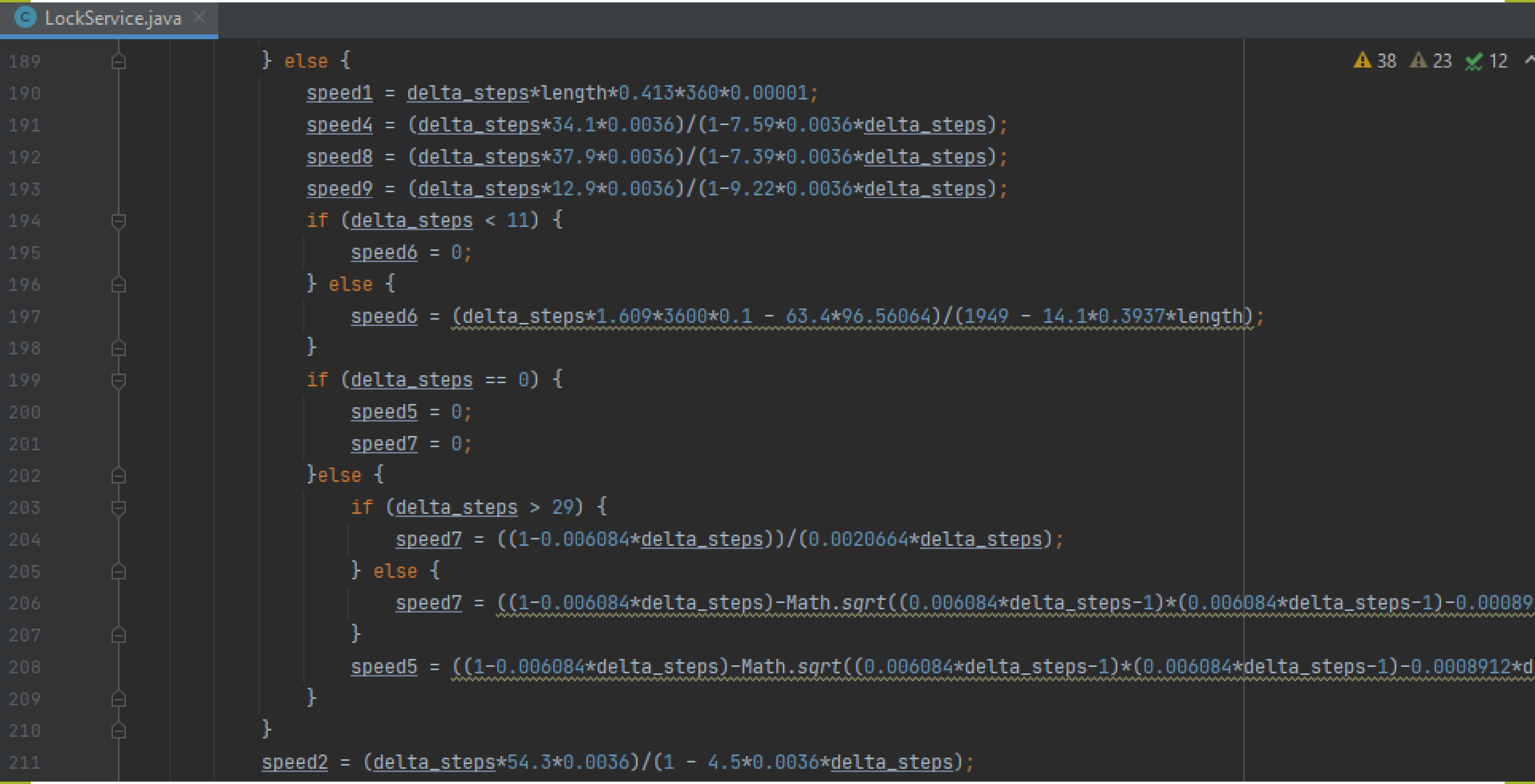
```java
    public double getSpeed(double length){
        double speed = 0;
        if (oldsteps != 0) {
            double delta_steps = steps - oldsteps;
            speed = (0.010872*delta_steps*length)/(20-delta_steps*length*9*0.0001);
            oldsteps = steps;
        } else {
            oldsteps = steps;
        }
        return speed;
    }
```



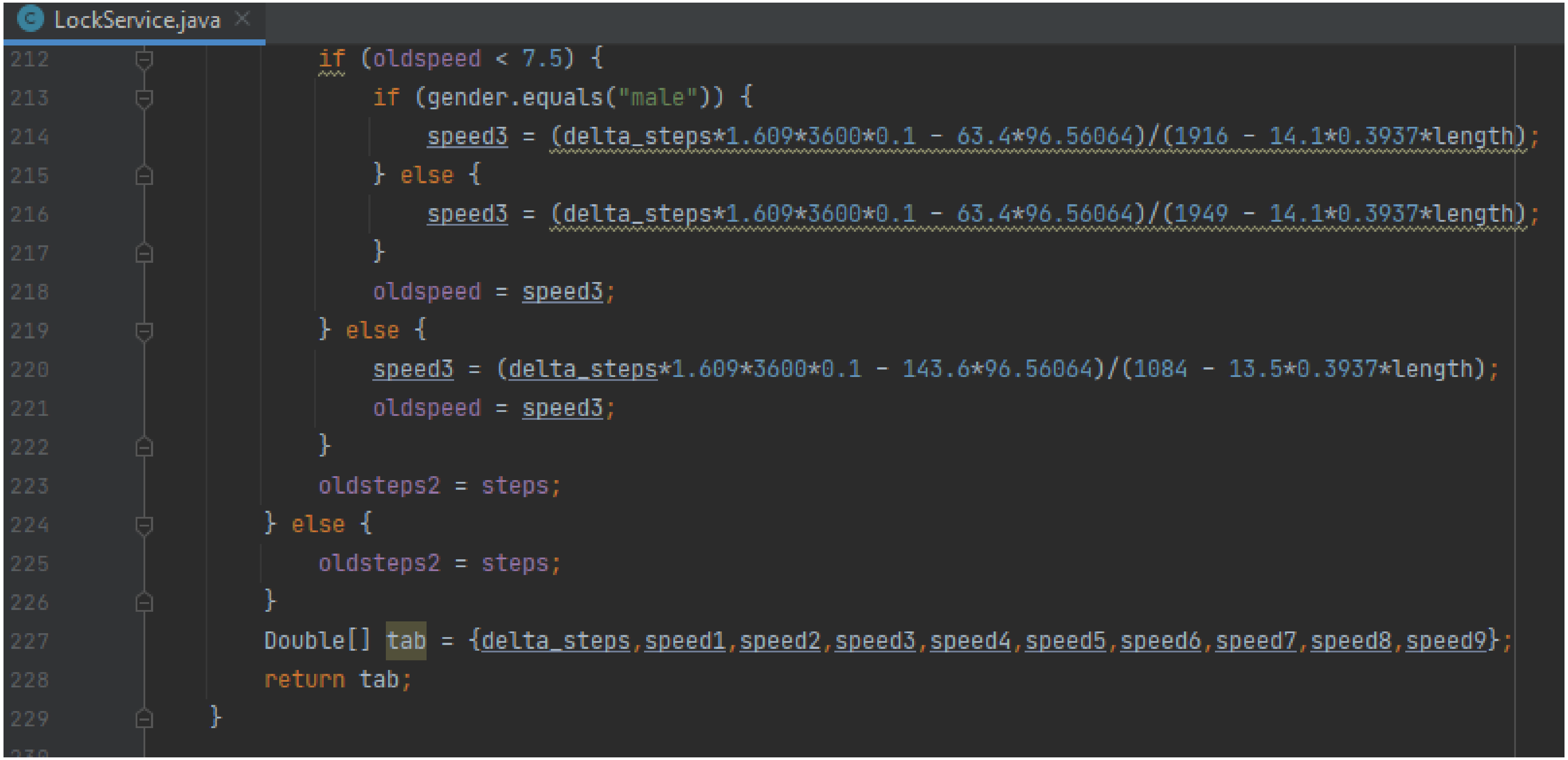
```java
    // TEST
    public Double[] getSpeedTest(double length, String gender){
        double speed1 = 0;
        double speed2 = 0;
        double speed3 = 0;
        double speed4 = 0;
        double speed5 = 0;
        double speed6 = 0;
        double speed7 = 0;
        double speed8 = 0;
        double speed9 = 0;
        double delta_steps = 0;
        if (oldsteps2 != 0) {
            delta_steps = steps - oldsteps2;
```



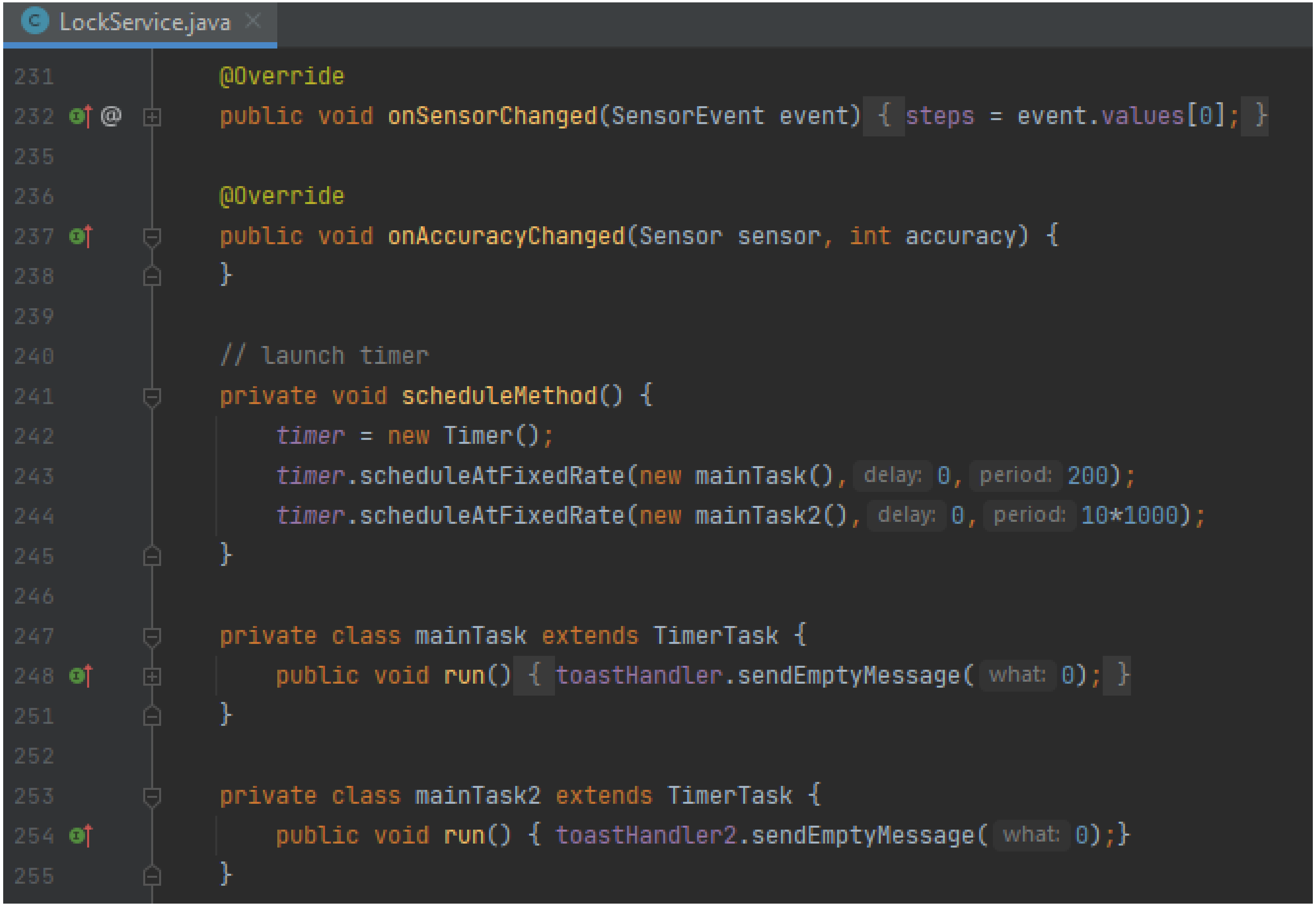
```java
if (gender.equals("male")) {
    speed1 = delta_steps*length*0.415*360*0.00001;
    speed4 = (delta_steps*33.3*0.0036)/(1-7.2*0.0036*delta_steps);
    speed8 = (delta_steps*37.4*0.0036)/(1-6.85*0.0036*delta_steps);
    speed9 = (delta_steps*16.4*0.0036)/(1-8.49*0.0036*delta_steps);
    if (delta_steps < 11) {
        speed6 = 0;
    } else {
        speed6 = (delta_steps*1.609*3600*0.1 - 63.4*96.56064)/(1916 - 14.1*0.3937*length);
    }
    if (delta_steps == 0) {
        speed5 = 0;
        speed7 = 0;
    }else {
        if (delta_steps > 29) {
            speed7 = ((1-0.00918*delta_steps))/(0.0016272*delta_steps);
        } else {
            speed7 = ((1-0.00918*delta_steps)-Math.sqrt((0.00918*delta_steps-1)*(0.00918*delta_steps-1)-0.00062914*de
        }
        speed5 = ((1-0.00918*delta_steps)-Math.sqrt((0.00918*delta_steps-1)*(0.00918*delta_steps-1)-0.00062914*delta_
    }
} else {
```



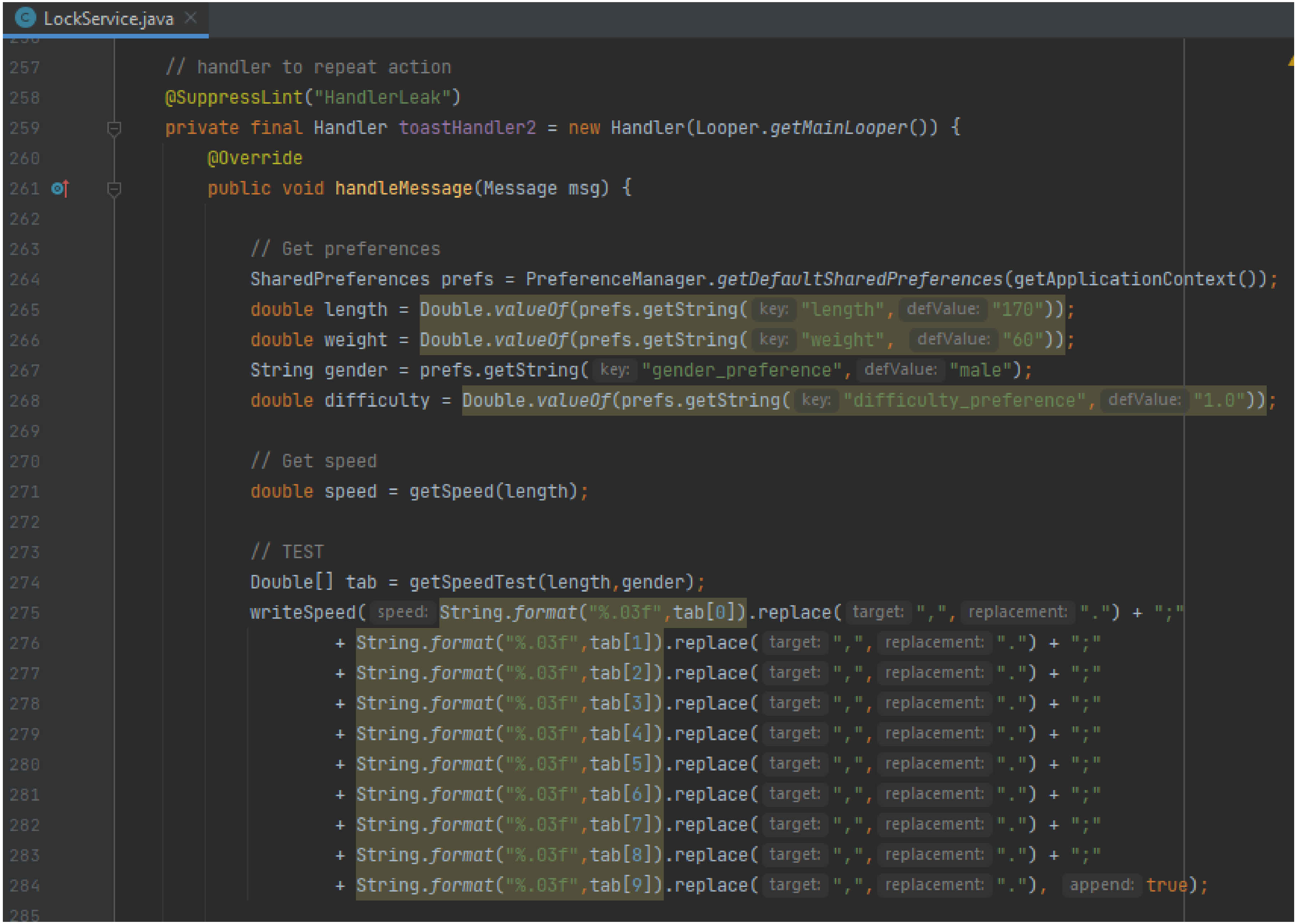
```java
} else {
    speed1 = delta_steps*length*0.413*360*0.00001;
    speed4 = (delta_steps*34.1*0.0036)/(1-7.59*0.0036*delta_steps);
    speed8 = (delta_steps*37.9*0.0036)/(1-7.39*0.0036*delta_steps);
    speed9 = (delta_steps*12.9*0.0036)/(1-9.22*0.0036*delta_steps);
    if (delta_steps < 11) {
        speed6 = 0;
    } else {
        speed6 = (delta_steps*1.609*3600*0.1 - 63.4*96.56064)/(1949 - 14.1*0.3937*length);
    }
    if (delta_steps == 0) {
        speed5 = 0;
        speed7 = 0;
    }else {
        if (delta_steps > 29) {
            speed7 = ((1-0.006084*delta_steps))/(0.0020664*delta_steps);
        } else {
            speed7 = ((1-0.006084*delta_steps)-Math.sqrt((0.006084*delta_steps-1)*(0.006084*delta_steps-1)-0.00089
        }
        speed5 = ((1-0.006084*delta_steps)-Math.sqrt((0.006084*delta_steps-1)*(0.006084*delta_steps-1)-0.0008912*d
    }
}
speed2 = (delta_steps*54.3*0.0036)/(1 - 4.5*0.0036*delta_steps);
```



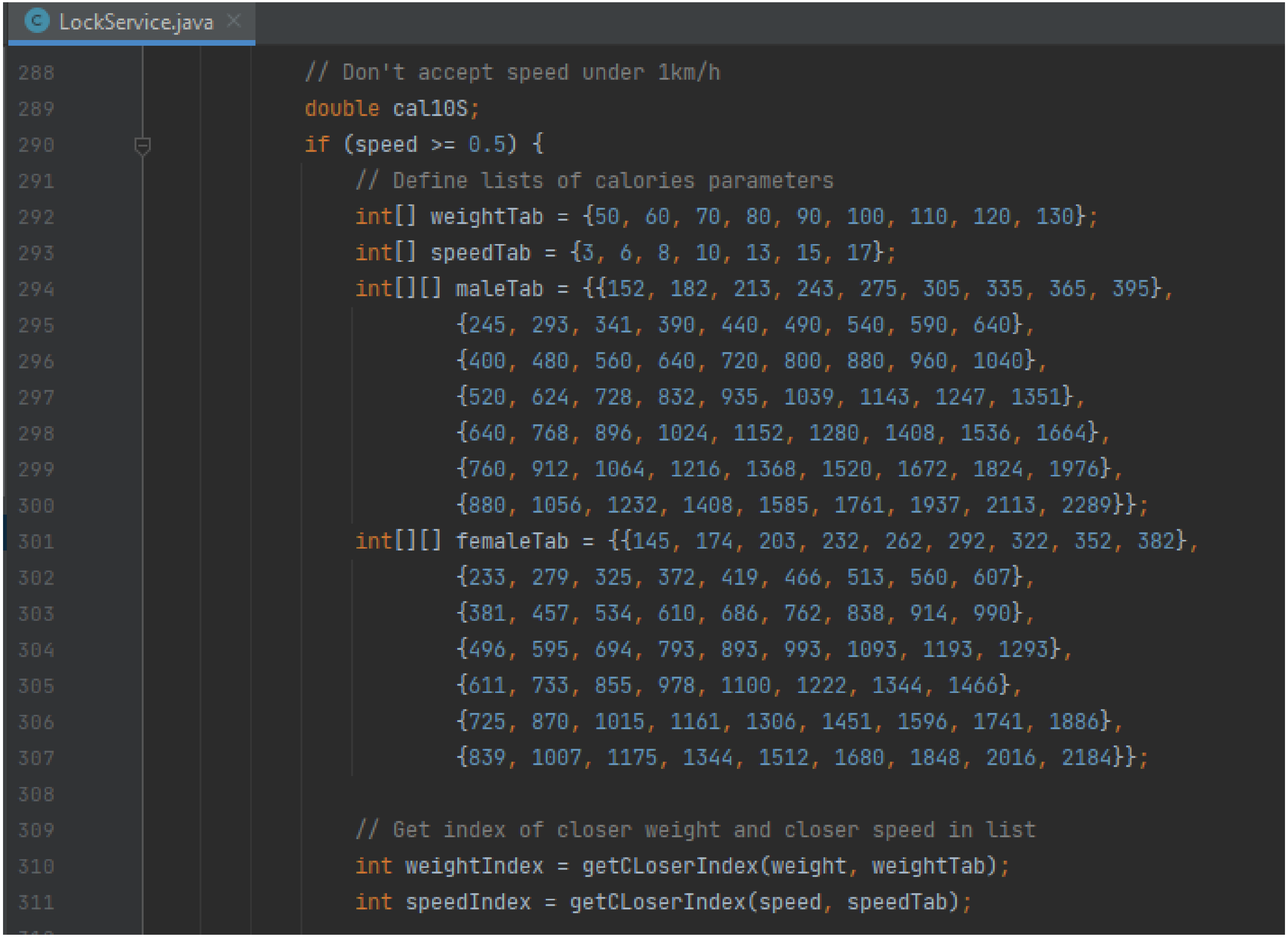
```java
            if (oldspeed < 7.5) {
                if (gender.equals("male")) {
                    speed3 = (delta_steps*1.609*3600*0.1 - 63.4*96.56064)/(1916 - 14.1*0.3937*length);
                } else {
                    speed3 = (delta_steps*1.609*3600*0.1 - 63.4*96.56064)/(1949 - 14.1*0.3937*length);
                }
                oldspeed = speed3;
            } else {
                speed3 = (delta_steps*1.609*3600*0.1 - 143.6*96.56064)/(1084 - 13.5*0.3937*length);
                oldspeed = speed3;
            }
            oldsteps2 = steps;
        } else {
            oldsteps2 = steps;
        }
        Double[] tab = {delta_steps,speed1,speed2,speed3,speed4,speed5,speed6,speed7,speed8,speed9};
        return tab;
    }
```



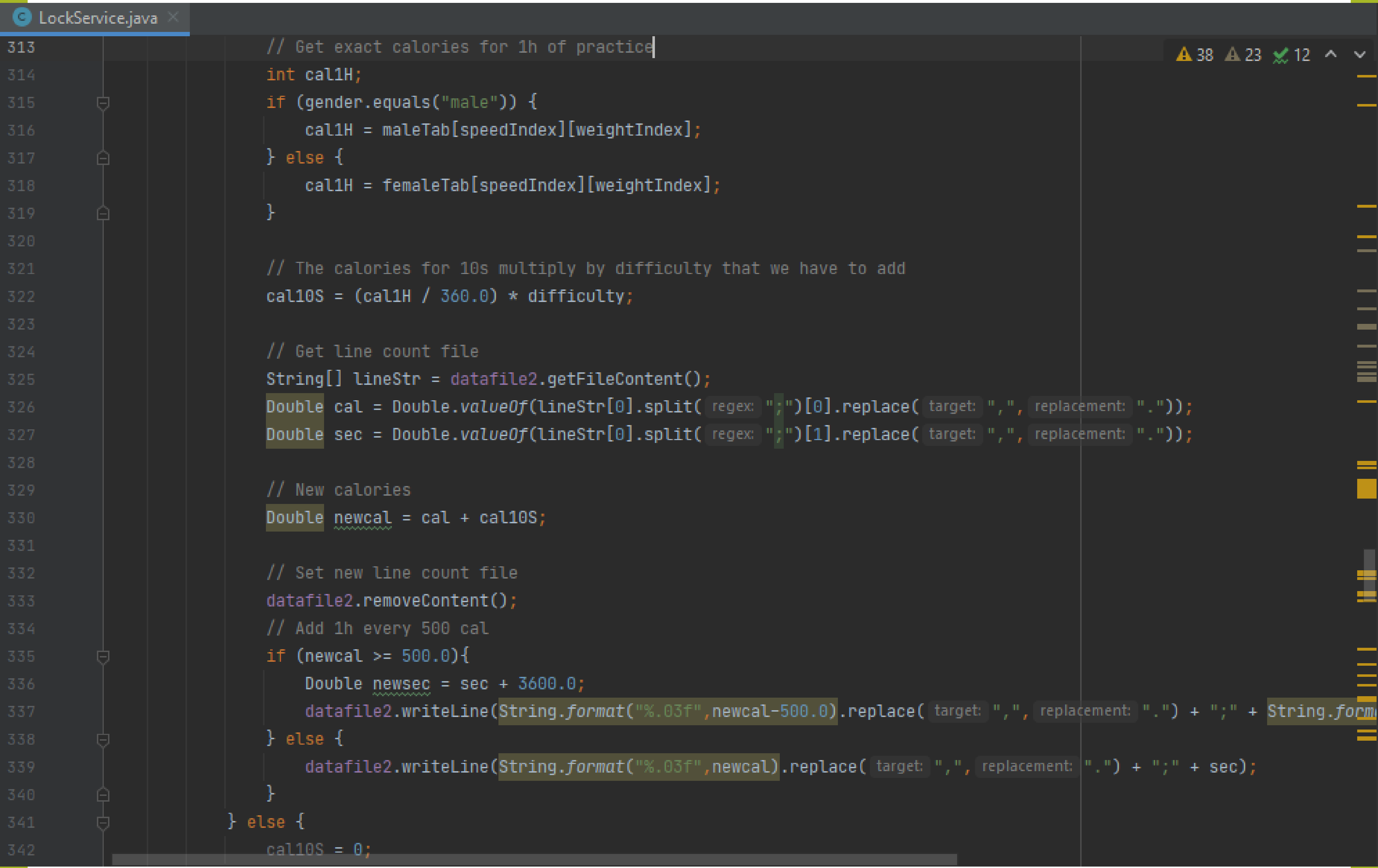
```java
    @Override
    public void onSensorChanged(SensorEvent event) { steps = event.values[0]; }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }

    // launch timer
    private void scheduleMethod() {
        timer = new Timer();
        timer.scheduleAtFixedRate(new mainTask(), delay: 0, period: 200);
        timer.scheduleAtFixedRate(new mainTask2(), delay: 0, period: 10*1000);
    }

    private class mainTask extends TimerTask {
        public void run() { toastHandler.sendEmptyMessage( what: 0); }
    }

    private class mainTask2 extends TimerTask {
        public void run() { toastHandler2.sendEmptyMessage( what: 0);}
    }
```



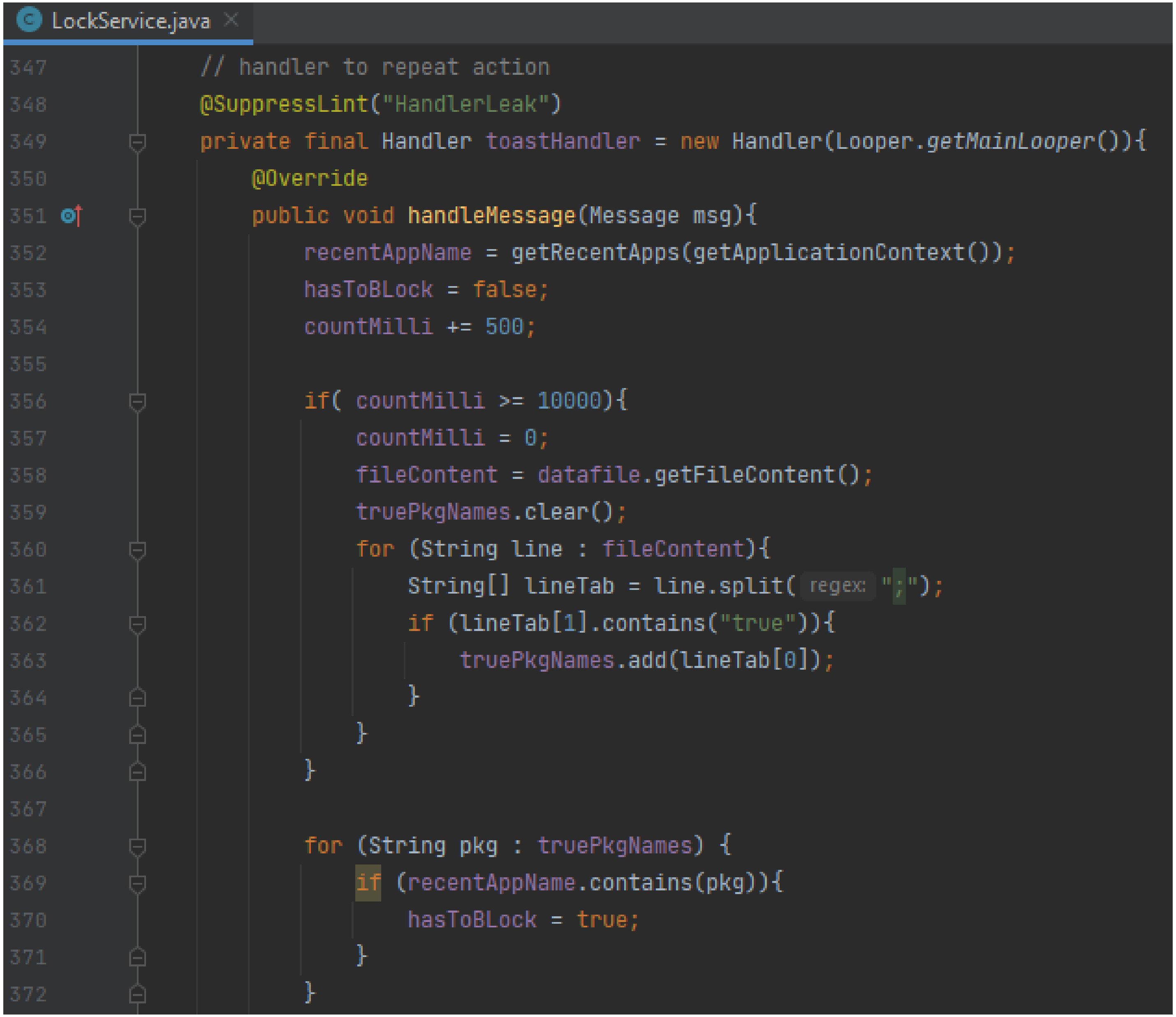
```java
    // handler to repeat action
    @SuppressLint("HandlerLeak")
    private final Handler toastHandler2 = new Handler(Looper.getMainLooper()) {
        @Override
        public void handleMessage(Message msg) {

            // Get preferences
            SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
            double length = Double.valueOf(prefs.getString( key: "length", defValue: "170"));
            double weight = Double.valueOf(prefs.getString( key: "weight",  defValue: "60"));
            String gender = prefs.getString( key: "gender_preference", defValue: "male");
            double difficulty = Double.valueOf(prefs.getString( key: "difficulty_preference", defValue: "1.0"));

            // Get speed
            double speed = getSpeed(length);

            // TEST
            Double[] tab = getSpeedTest(length,gender);
            writeSpeed( speed: String.format("%.03f",tab[0]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[1]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[2]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[3]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[4]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[5]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[6]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[7]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[8]).replace( target: ",", replacement: ".") + ";"
                    + String.format("%.03f",tab[9]).replace( target: ",", replacement: "."),  append: true);
```



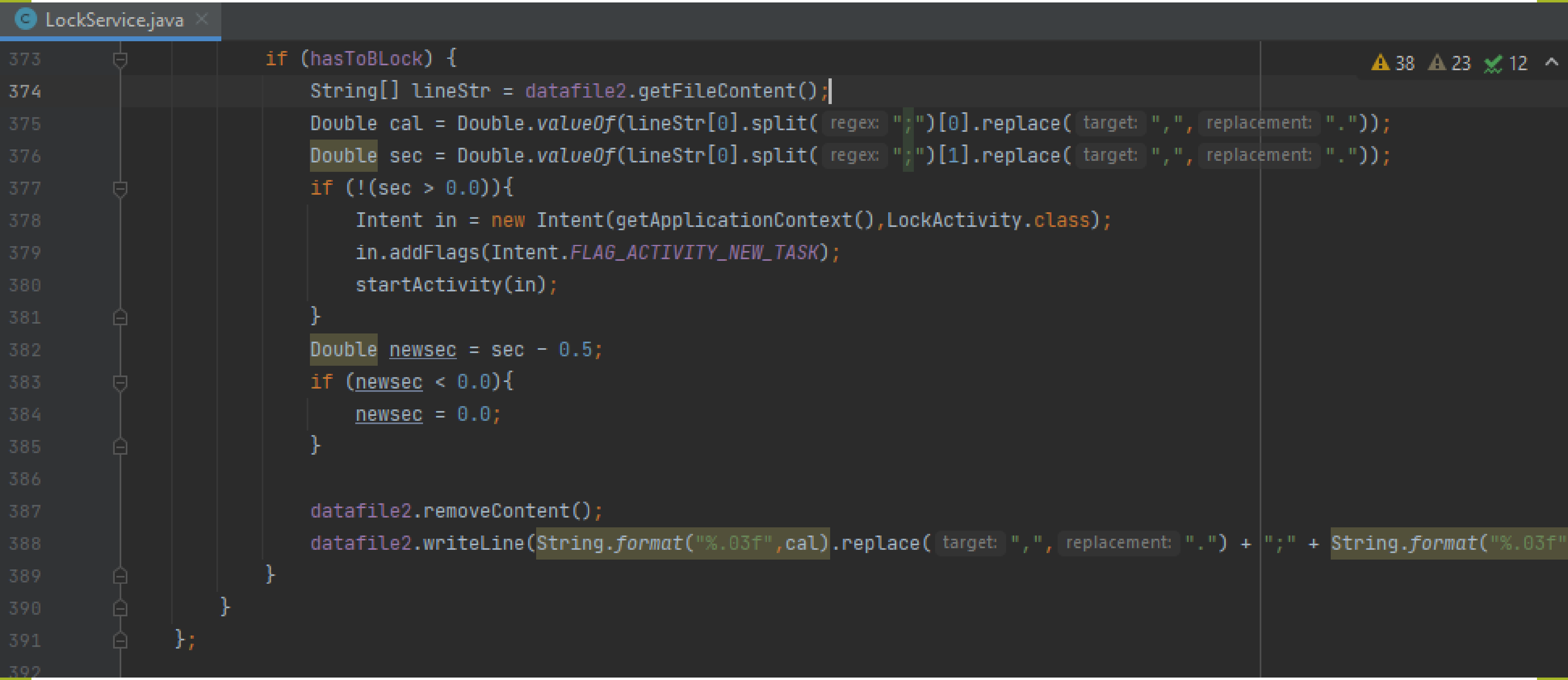
```java
        // Don't accept speed under 1km/h
        double cal10S;
        if (speed >= 0.5) {
            // Define lists of calories parameters
            int[] weightTab = {50, 60, 70, 80, 90, 100, 110, 120, 130};
            int[] speedTab = {3, 6, 8, 10, 13, 15, 17};
            int[][] maleTab = {{152, 182, 213, 243, 275, 305, 335, 365, 395},
                    {245, 293, 341, 390, 440, 490, 540, 590, 640},
                    {400, 480, 560, 640, 720, 800, 880, 960, 1040},
                    {520, 624, 728, 832, 935, 1039, 1143, 1247, 1351},
                    {640, 768, 896, 1024, 1152, 1280, 1408, 1536, 1664},
                    {760, 912, 1064, 1216, 1368, 1520, 1672, 1824, 1976},
                    {880, 1056, 1232, 1408, 1585, 1761, 1937, 2113, 2289}};
            int[][] femaleTab = {{145, 174, 203, 232, 262, 292, 322, 352, 382},
                    {233, 279, 325, 372, 419, 466, 513, 560, 607},
                    {381, 457, 534, 610, 686, 762, 838, 914, 990},
                    {496, 595, 694, 793, 893, 993, 1093, 1193, 1293},
                    {611, 733, 855, 978, 1100, 1222, 1344, 1466},
                    {725, 870, 1015, 1161, 1306, 1451, 1596, 1741, 1886},
                    {839, 1007, 1175, 1344, 1512, 1680, 1848, 2016, 2184}};

            // Get index of closer weight and closer speed in list
            int weightIndex = getCLoserIndex(weight, weightTab);
            int speedIndex = getCLoserIndex(speed, speedTab);
```



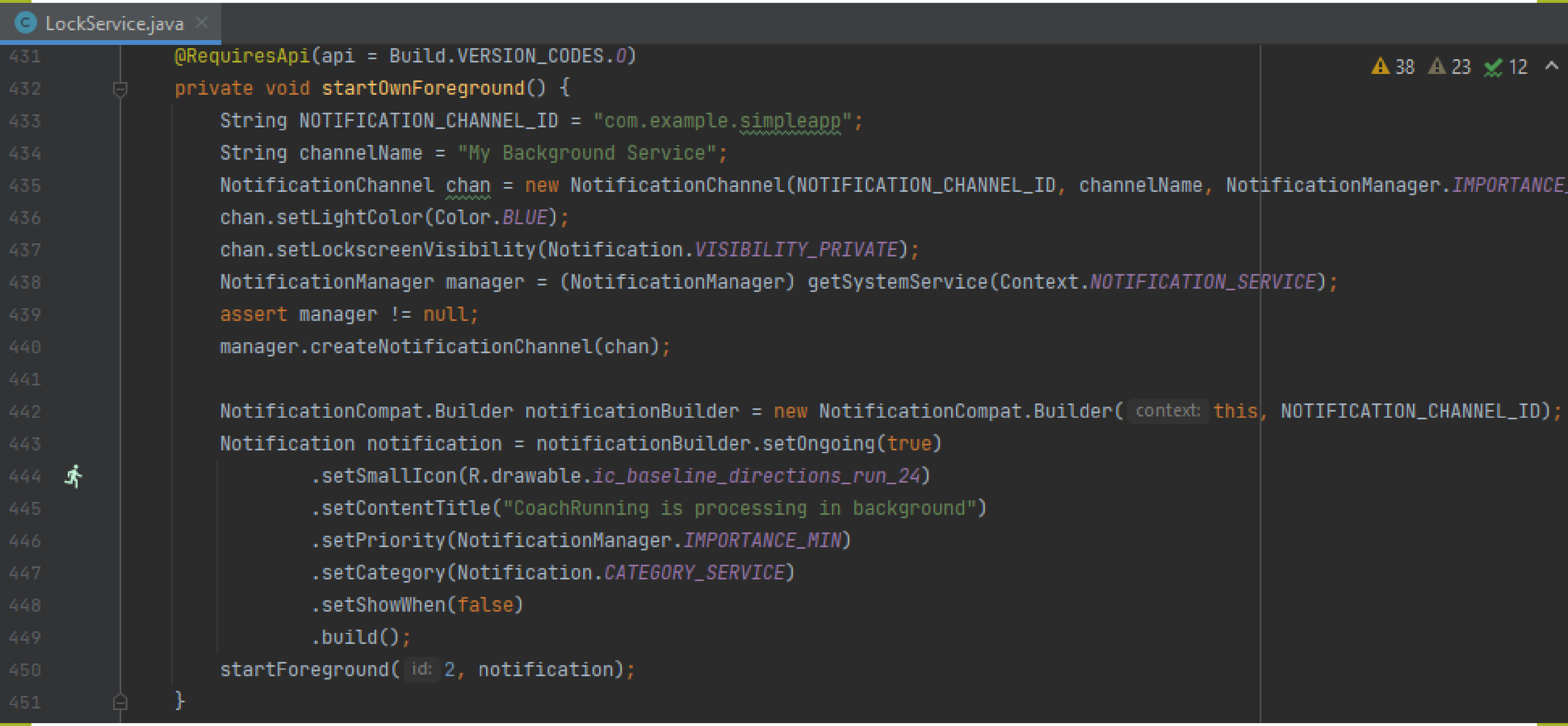
```java
                // Get exact calories for 1h of practice
                int cal1H;
                if (gender.equals("male")) {
                    cal1H = maleTab[speedIndex][weightIndex];
                } else {
                    cal1H = femaleTab[speedIndex][weightIndex];
                }

                // The calories for 10s multiply by difficulty that we have to add
                cal10S = (cal1H / 360.0) * difficulty;

                // Get line count file
                String[] lineStr = datafile2.getFileContent();
                Double cal = Double.valueOf(lineStr[0].split( regex: ";")[0].replace( target: ",", replacement: "."));
                Double sec = Double.valueOf(lineStr[0].split( regex: ";")[1].replace( target: ",", replacement: "."));

                // New calories
                Double newcal = cal + cal10S;

                // Set new line count file
                datafile2.removeContent();
                // Add 1h every 500 cal
                if (newcal >= 500.0){
                    Double newsec = sec + 3600.0;
                    datafile2.writeLine(String.format("%.03f",newcal-500.0).replace( target: ",", replacement: ".") + ";" + String.form
                } else {
                    datafile2.writeLine(String.format("%.03f",newcal).replace( target: ",", replacement: ".") + ";" + sec);
                }
            } else {
                cal10S = 0;
```



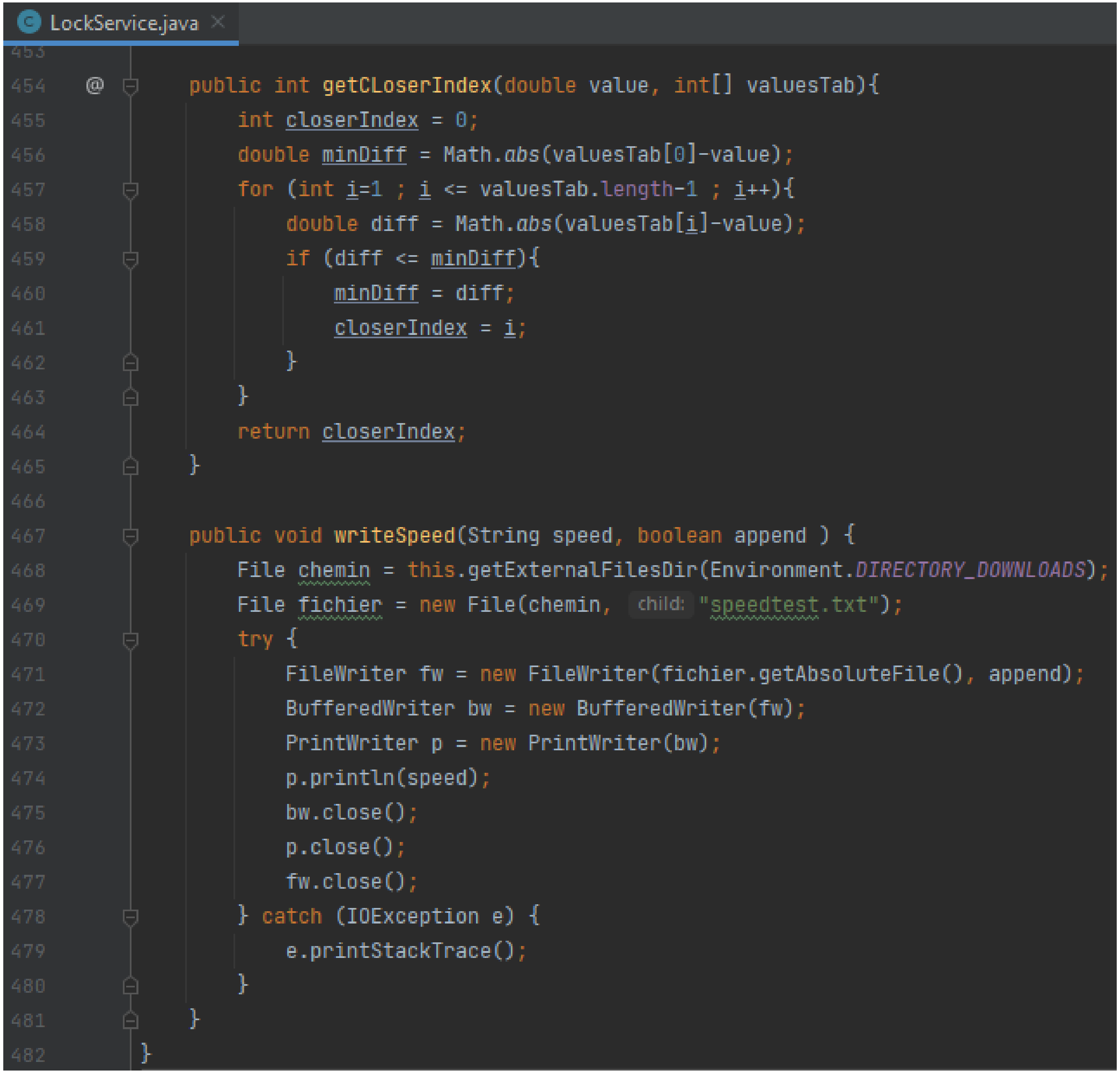
```java
    // handler to repeat action
    @SuppressLint("HandlerLeak")
    private final Handler toastHandler = new Handler(Looper.getMainLooper()){
        @Override
        public void handleMessage(Message msg){
            recentAppName = getRecentApps(getApplicationContext());
            hasToBLock = false;
            countMilli += 500;

            if( countMilli >= 10000){
                countMilli = 0;
                fileContent = datafile.getFileContent();
                truePkgNames.clear();
                for (String line : fileContent){
                    String[] lineTab = line.split(";");
                    if (lineTab[1].contains("true")){
                        truePkgNames.add(lineTab[0]);
                    }
                }
            }

            for (String pkg : truePkgNames) {
                if (recentAppName.contains(pkg)){
                    hasToBLock = true;
                }
            }
```



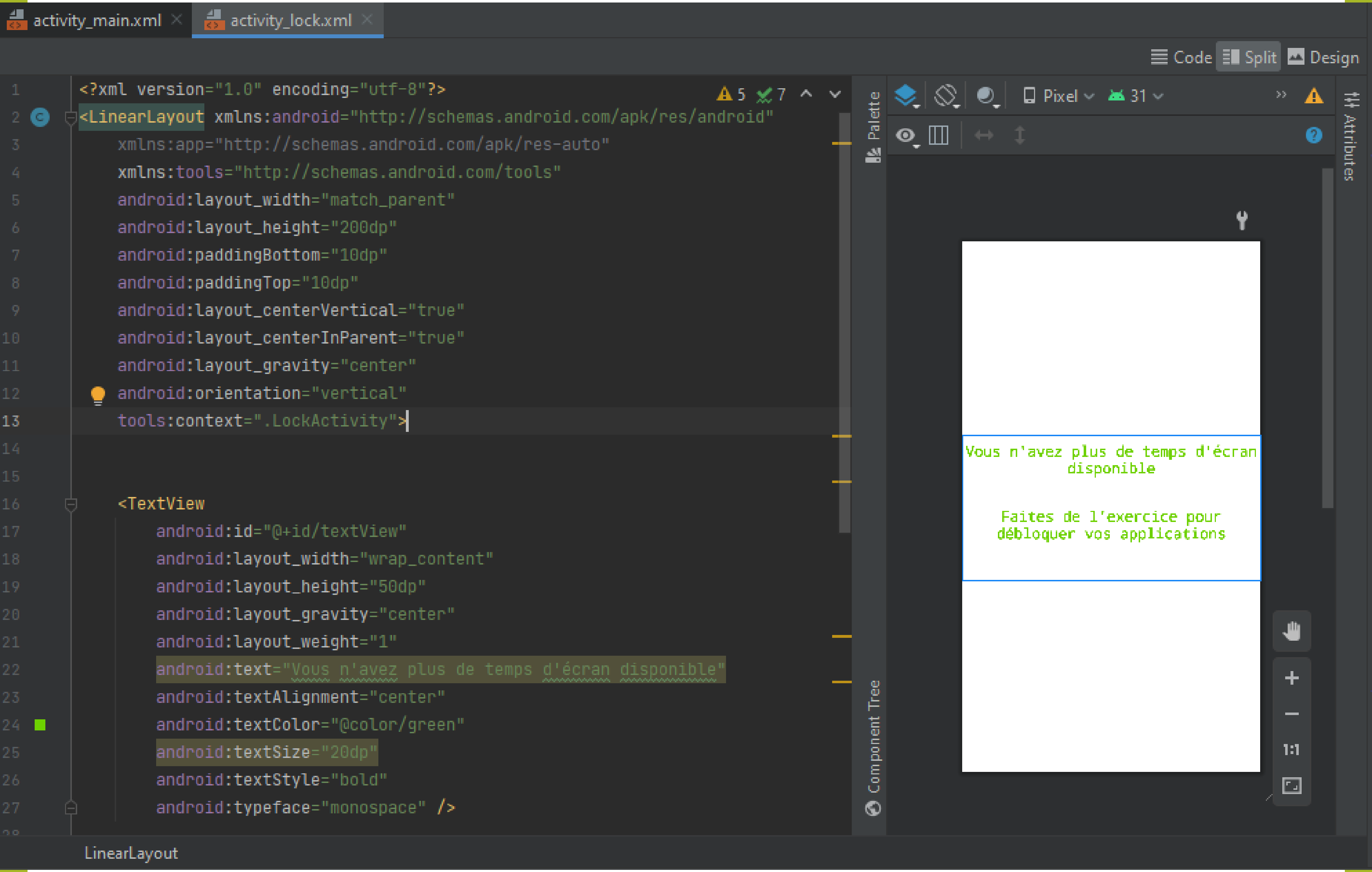
```
        if (hasToBLock) {
            String[] lineStr = datafile2.getFileContent();
            Double cal = Double.valueOf(lineStr[0].split( regex: ";")[0].replace( target: ",", replacement: "."));
            Double sec = Double.valueOf(lineStr[0].split( regex: ";")[1].replace( target: ",", replacement: "."));
            if (!(sec > 0.0)){
                Intent in = new Intent(getApplicationContext(),LockActivity.class);
                in.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(in);
            }
            Double newsec = sec - 0.5;
            if (newsec < 0.0){
                newsec = 0.0;
            }

            datafile2.removeContent();
            datafile2.writeLine(String.format("%.03f",cal).replace( target: ",", replacement: ".") + ";" + String.format("%.03f"
        }
    }
};
```



```java
    @RequiresApi(api = Build.VERSION_CODES.O)
    private void startOwnForeground() {
        String NOTIFICATION_CHANNEL_ID = "com.example.simpleapp";
        String channelName = "My Background Service";
        NotificationChannel chan = new NotificationChannel(NOTIFICATION_CHANNEL_ID, channelName, NotificationManager.IMPORTANCE_
        chan.setLightColor(Color.BLUE);
        chan.setLockscreenVisibility(Notification.VISIBILITY_PRIVATE);
        NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        assert manager != null;
        manager.createNotificationChannel(chan);

        NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder( context: this, NOTIFICATION_CHANNEL_ID);
        Notification notification = notificationBuilder.setOngoing(true)
                .setSmallIcon(R.drawable.ic_baseline_directions_run_24)
                .setContentTitle("CoachRunning is processing in background")
                .setPriority(NotificationManager.IMPORTANCE_MIN)
                .setCategory(Notification.CATEGORY_SERVICE)
                .setShowWhen(false)
                .build();
        startForeground( id: 2, notification);
    }
```



```java
    public int getCLoserIndex(double value, int[] valuesTab){
        int closerIndex = 0;
        double minDiff = Math.abs(valuesTab[0]-value);
        for (int i=1 ; i <= valuesTab.length-1 ; i++){
            double diff = Math.abs(valuesTab[i]-value);
            if (diff <= minDiff){
                minDiff = diff;
                closerIndex = i;
            }
        }
        return closerIndex;
    }

    public void writeSpeed(String speed, boolean append ) {
        File chemin = this.getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS);
        File fichier = new File(chemin, child: "speedtest.txt");
        try {
            FileWriter fw = new FileWriter(fichier.getAbsoluteFile(), append);
            BufferedWriter bw = new BufferedWriter(fw);
            PrintWriter p = new PrintWriter(bw);
            p.println(speed);
            bw.close();
            p.close();
            fw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:layout_centerVertical="true"
    android:layout_centerInParent="true"
    android:layout_gravity="center"
    android:orientation="vertical"
    tools:context=".LockActivity">


    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:text="Vous n'avez plus de temps d'écran disponible"
        android:textAlignment="center"
        android:textColor="@color/green"
        android:textSize="20dp"
        android:textStyle="bold"
        android:typeface="monospace" />
```

Vous n'avez plus de temps d'écran disponible

Faites de l'exercice pour débloquer vos applications