

Компилятор Java и IDE: взаимовыгодное сотрудничество

Чашников Николай, JetBrains

Компиляция java-файла – это просто

```
HelloWorld.java
```

```
> javac HelloWorld.java
```

```
HelloWorld.class
```

Даже если их несколько

```
> javac *.java
```

```
> mvn compile
```

```
> ant -f compile.xml
```

ПРОГРАММЕРСКАЯ ОТМАЗКА #1 ДЛЯ
ЗАКОННОГО ОТЛЫНИВАНИЯ ОТ РАБОТЫ:

«МОЙ КОД КОМПИЛИРУЕТСЯ».



I'm working on a big project and am having a really difficult time getting IntelliJ to build it. Sometimes it finishes after 20 or so minutes, sometimes never.

I know this is an issue for my entire company, but people have become complacent and accept that their IDE will remain locked for hours.

Build is extremely slow

The compilation takes ~4 seconds to complete

Инкрементальная компиляция

Перекомпилировать изменившиеся файлы
и некоторые другие

Добавление поля в класс

```
public class Cat {  
}
```



```
public class Cat {  
    protected Tail tail;  
}
```

Что надо перекомпилировать, кроме Cat.java?

- A. ничего
- B. классы, в которых используется Cat
- C. наследников класса Cat
- D. что-то другое

Добавление поля в класс

...

```
Tail tail = new Tail();
```

```
new Cat() {
```

```
{
```

```
    tail.pull();
```

```
}
```

```
};
```

...

Добавление `private` поля в класс

```
public class Cat
    extends Animal {
}
```



```
public class Cat
    extends Animal {
    private Tail tail;
}
```

Что надо
перекомпилировать, кроме
Cat.java?

- A. ничего
- B. классы, в которых
используется Cat
- C. наследников класса Cat
- D. что-то другое

Добавление `private` поля в класс

```
public class Animal {  
    public Tail tail;  
}
```

...

```
Cat cat = new Cat();  
cat.tail.pull();
```

...

Изменение значения поля

```
public class Answers {  
    public static  
        int ANSWER = 42;  
}
```



```
public class Answers {  
    public static  
        int ANSWER = 43;  
}
```

Что надо перекомпилировать,
кроме Answers.java?

- A. ничего
- B. классы, в которых
используется ANSWER
- C. наследников класса
Answers
- D. что-то другое

Изменение значения `static final` поля

```
public class Answers {  
    public static final  
        int ANSWER = 42;  
}
```



```
public class Answers {  
    public static final  
        int ANSWER = 43;  
}
```

Что надо перекомпилировать, кроме `Answers.java`?

- A. ничего
- B. классы, в которых используется `ANSWER`
- C. наследников класса `Answers`
- D. что-то другое

Изменение значения `static final` поля

```
public class Questions {  
    public int getAnswer() {  
        return Answers.ANSWER;  
    }  
}
```

Компилятор и IDE

Компилятор:

parsing

error reporting

byte-code generation

IDE:

parsing

error reporting

quick fixes

completion

refactoring

...

Eclipse

- собственный компилятор (ECSJ)
- запускается прямо в процессе IDE
- IDE переиспользует код компилятора
- компилятор запускается при сохранении файлов
- компилятор сам собирает информацию для инкрементальной перекомпиляции

NetBeans

- использует javac
- запускается прямо в процессе IDE
- IDE переиспользует код компилятора
- компилятор постоянно запускается в фоне
- для инкрементальной компиляции используется информация из IDE

IntelliJ IDEA

- может работать как с javac, так и с ECJ
- использует компилятор как чёрный ящик
- запускает его в отдельном процессе
- собственный парсер и анализатор в IDE
- для инкрементальной компиляции используется информация из IDE

Скомпилируется ли этот код?

```
class Root {}
```

```
class Inc<Z> {}
```

```
class C0<X> extends Inc<Inc<? super X>> {}
```

```
class Test {
```

```
    Inc<? super C0<Root>> cast(C0<Inc<? super Root>> c) {
```

```
        return c;
```

```
    }
```

```
}
```

А ЭТОТ?

```
class Root {}  
class Inc<Z> {}  
class C0<X> extends Inc<Inc<? super X>> {}  
class C1<X> extends C0<C0<X>> {}  
class Test {  
    Inc<? super C1<Root>> cast(C1<Inc<? super Root>> c) {  
        return c;  
    }  
}
```

Вот этот уже нет

```
class Root {}
class Inc<Z> {}
class C0<X> extends Inc<Inc<? super X>> {}
class C1<X> extends C0<C0<X>> {}
class C2<X> extends C1<C1<X>> {}
class C3<X> extends C2<C2<X>> {}
class C4<X> extends C3<C3<X>> {}
class C5<X> extends C4<C4<X>> {}
class C6<X> extends C5<C5<X>> {}
class C7<X> extends C6<C6<X>> {}
class C8<X> extends C7<C7<X>> {}
class C9<X> extends C8<C8<X>> {}
class Test {
    Inc<? super C9<Root>> cast(C9<Inc<? super Root>> c) {
        return c;
    }
}
```