

A non-structural-match type was used as the type of a const generic parameter.

Erroneous code example:

```
#![feature(adt_const_params)]
```

```
struct A;
```

```
struct B<const X: A>; // error!
```

Only structural-match types (that is, types that derive `PartialEq` and `Eq`) may be used as the types of const generic parameters.

To fix the previous code example, we derive `PartialEq` and `Eq`:

```
#![feature(adt_const_params)]
```

```
#[derive(PartialEq, Eq)] // We derive both traits here.
```

```
struct A;
```

```
struct B<const X: A>; // ok!
```